Machine Learning Techniques in Usage-Based Insurance: Use of Telematic Data in Auto Insurance

Helia Alipanah

A Thesis in

The Department of

Mathematics and Statistics

Presented in Partial Fulfilment of the Requirements

for the Degree of Master of Science (Mathematics) at

Concordia University

Montreal, Quebec, Canada

July 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis

prepared By: **Helia Alipanah**

Entitled: **Machine Learning Techniques in Usage-Based Insurance: Use of Telematic Data in Auto Insurance**

and submitted in partial fulfilment of the requirements for the degree of

Master of Science (Mathematics)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final Examining Committee:

______________________________ Examiner

Dr. Mélina Mailhot

______________________________ Examiner

______________________________ Thesis Co-Supervisor

Dr. J. Garrido

______________________________ Thesis Co-Supervisor

Dr. F. Godin

Approved by ______________________________

Chair of Department or Graduate Program Director

____

Dean of Faculty

Date ______________________________

Abstract

Machine Learning Techniques in Usage-based Insurance: Use of Telematic Data in Auto Insurance

by Helia Alipanah

The development of big data technologies and in-vehicle devices has contributed to the growth of Usage-Based Insurance (UBI) in recent years. These in-vehicle devices, such as GPS and sensors, collect certain variables that can represent the driving behaviour of policyholders. This collected data, called telematic data, consist of several variables that have strong relationship with the likelihood of having an accident. Consequently, one can use telematic data to improve risk assessments and personalize car insurance premiums.

In this thesis, a synthetic car insurance dataset emulated from a Canadian-based insurance company is used to investigate the use of telematic data in predicting the likelihood of having an accident. More precisely four machine learning techniques—logistic regression, random forests, gradient boosting trees, and feed-forward neural networks—are employed to predict the risk of having an accident. Actuaries often use white box machine learning methods like logistic regression for risk assessment due to their interpretability. However, these method are unable to detect non-linear relationships between variables accurately. Therefore, more complex machine learning techniques such as random forests, gradient boosting trees, and feed-forward neural networks are used to achieve more accurate risk assessment for accidents.

In addition, two variable importance assessment methods—Shapley decomposition and marginal performance loss upon feature removal—are employed to provide insights into the feature contributions in the overall predictive performance of the models.

Acknowledgments

I would like to express my heartfelt gratitude to my supervisors, Dr. José Garrido, and Dr. Frédéric Godin, for their guidance, support, and invaluable expertise throughout the entire process of this thesis. Their commitment, patience, and insightful feedback have been instrumental in shaping the direction and quality of this research. I am truly grateful for their mentorship and for instilling in me a passion for academic inquiry.

I would also like to extend my deepest appreciation to my husband, whose unwavering love, encouragement, and understanding have been my rock during this challenging journey. His belief in me and constant motivation have given me the strength to overcome obstacles and pursue my academic aspirations.

Furthermore, I am profoundly grateful to my parents for their endless love, unwavering support, and sacrifices they have made throughout my academic pursuit. Their unwavering belief in my abilities has been a constant source of inspiration and motivation. I am forever indebted to them for instilling in me the values of perseverance, determination, and the importance of education.

Lastly, I would like to extend my sincere appreciation to all my friends and family members who have provided me with encouragement, understanding, and support during this arduous yet rewarding endeavor.

Without the collective support and encouragement of these individuals, this thesis would not have been possible. Their contributions have shaped not only my academic journey but also my personal growth. I am truly grateful for their presence in my life.

Contents

List of Figures

List of Tables

1. *Introduction*

Automobile insurance companies traditionally determine the policyholders' premium by assessing their personal profile, aiming to establish fair rates that align with the individual's risk of accidents. These companies typically employ traditional factors to evaluate accident risk. These factors include personal-specific attributes, such as the driver's age, gender, driving experience, claims history, and city of residence. Additionally, car-specific factors, including the vehicle's model, age, and estimated annual mileage, also play a role in this assessment process. However, with advancements in technology and the collection of telematic data, the automobile insurance industry has undergone a transformative shift. Telematic data refers to the information collected by the devices installed in vehicles or sensors and GPS technology. Telematic data includes crucial details such as the vehicle's location, speed, acceleration, braking, engine performance, and other relevant metrics. Telematic data provide valuable insights into the driver behavior and driving patterns. Therefore, it has facilitated the introduction of Usage-Based Insurance (UBI), also known as pay-as-you-drive or pay-how-you-drive. By monitoring driving behavior and usage patterns, insurance companies can now assess the risk of accidents associated with individual policyholders more accurately. Safe drivers may be eligible for discounted premiums, while those with riskier driving habits may face higher rates. This shift from traditional risk assessment models to individualized pricing has brought greater fairness and accuracy to insurance pricing.

1.1 Research objectives

This research aims to address several key questions regarding the impact of telematic data on the assessment of the risk of accident in the automobile insurance industry. The primary objective of this research is to examine the impact of telematic data on the assessment of accident risk in the automobile insurance industry. By analyzing information collected through telematic devices, such as acceleration, braking, distance driven, cornering patterns, and other relevant metrics, insurers can gain a more comprehensive understanding of individual driving behaviors. Consequently, the risk of accident associated with policyholders can be assessed more accurately. Through a comparative analysis, this study will ascertain whether the use of telematic data improves the accuracy of risk assessment compared to traditional data alone. Moreover, this study explores the possibility of predicting accident risk more accurately by combining traditional data with telematic data. By using both data sources, insurers can potentially develop more robust predictive models.

Furthermore, this study will evaluate alternative machine learning methods, such as logistic regression, random forests, gradient boosting trees, and feed-forward neural networks, to predict accident risk. Actuaries traditionally use machine learning methods, such as generalized additive models (GAM) and logistic regression, to predict the risk of accident. These methods are easy to interpret and provide insight regarding factors affecting the risk of accident. However, advanced techniques like random forests, gradient boosting trees, and neural networks present the opportunity to capture complex relationships and non-linearities present in the data. This study analyzes whether these machine learning methods yield more accurate results.

1.2 Limitations

It is crucial to acknowledge certain limitations that may impact the generalizability of the findings. Firstly, the availability of data sources, particularly telematic data sources, poses a significant challenge. Insurance companies typically do not readily share their datasets due to privacy and proprietary concerns. Therefore, obtaining a comprehensive and suitable dataset for analysis can be challenging. The limited accessibility to telematic data may restrict the scope and sample size of the study, potentially influencing the generalizability of the results. In this study, a synthetic dataset emulated from a real dataset to address this limitation.

Secondly, the high-dimensional nature of telematic datasets demands careful preprocessing of the data. Feature selection and dimensionality reduction techniques may be required to handle the complexity of calculations, especially when applying machine learning methods. The preprocessing of data may introduce biases or impact the performance of the models.

Lastly, the imbalanced nature of the automobile insurance dataset presents another challenge. Insurance claims data often exhibit a significant class imbalance, with a few observations having claims compared to a majority of non-claim instances. This imbalance can affect the performance of predictive models and may require the use of specialized techniques, such as oversampling or undersampling, to mitigate the bias towards the majority class.

Despite these limitations, this study aims to provide valuable insights into the potential benefits of using telematic data for risk assessment and predictive modeling in the automobile insurance industry. The findings will contribute to the understanding of the effectiveness of using telematic data into risk assessment. Therefore, it assists insurance companies in making informed decisions regarding pricing strategies and encouraging safer driving behaviors among policyholders. Additionally, the identified limitations will highlight areas for further research. Moreover it highlights the need for collaborative efforts to overcome data availability and processing challenges in future studies.

1.3 Literature review

Telematic data, collected through GPS, smartphones, sensors, and embedded equipment, provides information such as distance traveled, speed, acceleration, hard brakings, and location. These features offer insights into the driving habits of the insured. The introduction of telematic data has significantly improved premium calculations in the insurance industry. Insurers are now able to determine the risk of accidents more accurately based on the driving behavior of policyholders (Qi et al., 2018).

The introduction of telematic data and the development of usage-based insurance (UBI) have allowed for a more accurate assessment of risk, based on driving patterns. UBI started with a simple pay-as-you-drive (PAYD) model, which calculates premiums solely based on distance driven without distinguishing between safe drivers and risky drivers. Then UBI developed to more sophisticating approaches like pay-how-you-drive (PHYD) and manage-how-you-drive (MHYD) (Arumugam and Bhargavi (2019)). PHYD calculate the premiums based on driving behaviours such as excess speed, hard acceleration, and hard braking, while MHYD goes a step further by providing real-time alerts to help drivers reduce the risk of accident.

PHYD and MHYD provide a linkage between the insurance premiums and driving behaviour, which can serve as a potential mechanism for reducing risky driving behaviours and improving road safety. The findings in Bolderdijk et al. (2011) show that policyholders who participated in PAYD insurance exhibit lower levels of speeding comparing to policyholders with fixed premiums. This suggests that the implementation of PAYD insurance had a positive effect on reducing speeding behavior among young drivers.

Numerous studies have shown that predictive models incorporating telematic data outperform those based solely on traditional covariates when assessing the risk of accident (Fan and Wang (2017); Gao et al. (2019a); Barry and Charpentier (2020)). However, the most significant improvements are observed when both telematic and traditional data sources are included in the model, as they capture different aspects of risk and complement each other effectively (Baecke and Bocca, 2017).

Dealing with telematics datasets can be challenging due to their high dimensionality, specially when using machine learning methods. Using high dimensional dataset in machine learning requires complex calculations which are usually time consuming. Moreover, as the dimension of a dataset increases, the feature space grows exponentially, which lead to sparsity, and make it difficult for machine learning techniques to find meaningful patterns and relationships. To address this problem, researchers have employed techniques such as heatmaps to extract covariate information from telematic data (Gao et al. (2019b); Gao et al. (2021); Wüthrich (2017)). Speed and acceleration are parameters that affect the risk of accident significantly. Therefore, one can present these two features by a speed-acceleration (v-a) heatmap, which shows the time spent in a specific speed and acceleration state for a given driver. Then, one can extract covariates from the v-a heatmap by using the K-medoid algorithm and principal component analysis. The K-medoid method is similar to k-means, but any function can be used for measuring distance. Therefore, it is more robust.

Various machine learning techniques have been explored for telematic data analysis. Boucher et al. (2017) investigate the effect of the distance traveled and the exposure time on the risk of accident. The visualization of data reflects a non-linear relationship between the number of claims and the distance traveled. They explain the non-linearity by the fact that drivers with more experience exhibit less risk of accident. A Poisson generalized additive model (GAM) based on independent cubic splines is employed to capture this non-linear relationship. In another study, Boucher and Turcotte (2020) argued that additional kilometers of driving might not significantly reduce accident risk due to the saturation of driving experience, meaning that most of policyholders have enough driving experience and a few additional kilometers of driving do not add sufficient experience to reduce the risk of accident further. The surprising pattern in risk of accident can be caused by the residual individual heterogeneity that the basic Poisson GAM is not able to capture because of the independence of the observations (the observations are not independent as the same policyholder can be observed over many contracts). To address this, they extended the basic Poisson GAM model, relaxing the assumption of observation independence. Boucher et al. (2013) also investigate the impact of the distance driven on the risk of accident. The authors explain the nonlinear relationship with a generalization of the offset Poisson regression, such that for every observation x_i, we have $\lambda_i = \exp(x_i\beta + c \times \log(km))$. This model perfectly matches the observed data.

In terms of data mining techniques, researcher have explored various machine learning algorithms to analyze telematic data and improve risk assessment. Baecke and Bocca (2017) compared the performance of logistic regression, random forests, and one-dimensional feed-forward neural networks in assessing risk using telematic data. The results showed that logistic regression and fee-forward neural networks outperformed random forest, with feed-forward neural networks demonstrating the best predictive performance. The ability of feed-forward neural networks to capture complex patterns and relationships in the data contributed to their superior performance in risk assessment.

Gao and Wüthrich (2019) used high-frequency GPS-collected telematic data and focused on study individual trips by analyzing their time series of speed, acceleration, braking and change in angles. They trained a deep convolutional neural network (CNN) to identify the driver of each trip based on their time-series patterns. The CNN's ability to extract and analyze sequential data enabled accurate identification of individual drivers, demonstrating the potential of deep learning techniques in telematic data analysis.

Huang and Meng (2019) investigated the extraction of significant variables and the prediction of claim frequency using various machine learning models, including logistic regression, support vector machines, random forests, XGBoost, and artificial neural networks. Their study demonstrated that advanced machine learning techniques achieved good performance in predicting claim frequency, with the XGBoost model exhibiting the highest accuracy. The ability of XGBoost to handle complex interactions and nonlinear relationships in the data contributed to its superior predictive power.

Gao et al. (2021) analyzed the telematic data by presenting data in a speed-acceleration (v-a) heatmap. They established a densely connected feed-forward neural network and a convolutional neural network (CNN) to process the telematic data and extract relevant covariates. Both methods yielded similar results in risk assessment, but the CNN, with its ability to capture spatial patterns, proved to be more interpretable and used fewer parameters than the feed-forward neural network.

Meng et al. (2021) converted the telematic data into time series and applied a one-dimensional convolutional neural network (CNN) to classify each trip into safe or dangerous categories. Their study showed significant improvements in prediction performance compared to the Poisson generalized linear model (GLM), highlighting the effectiveness of CNNs in capturing temporal patterns and identifying risky driving behaviour.

Yu et al. (2021) applied a three-layer backpropagation (BP) neural network to estimate the total claim amount. The author uses a genetic algorithm to optimize the network's parameters, aiming to improve convergence speed and find the global optimum. The research shows that this method can predict claims more accurately comparing to the common-used methods such as GAM.

These studies collectively demonstrate the potential of various machine learning techniques, such as feed-forward neural networks, convolutional neural networks, and advanced models like XGBoost, in analyzing telematic data and improving risk assessment in automobile insurance. By leveraging these techniques, insurers can gain deeper insights into driver behavior, identify risk factors more accurately, and make more informed decisions regarding policy premiums and coverage.

Car insurance datasets contain usually a large number of zero claims. Applying a machine learning method on a unbalanced dataset is challenging, and it affects the prediction accuracy. To eliminate the effect of large number of zeros and to assess the risk of accident, one can use a zero-inflated regression which assumes that with a certain probability, the only possible observation is zero, and with the remaining probability, a random variable from a specific distribution is observed. In the context of telematics data, Sun et al. (2021) use Poisson zero-inflated regression and negative binomial zero-inflated regression models to assess the risk of accidents. These models take into account the counts of excess speed, high-speed braking, hard acceleration, and deceleration as predictors for estimating the risk of accidents. The study compares the performance of the Poisson and negative binomial regression models, with the latter demonstrating better predictive capabilities.

Guillen et al. (2021) employed a zero-inflated Poisson regression model to assess the risk of accidents. This approach assumes that with a certain probability, the only possible observation is zero, while with the remaining probability, a random variable with a specific distribution is observed. By incorporating this mixture model, insurers can account for the excess zeros in the dataset and improve the accuracy of risk assessment.

Another approach to mitigating the impact of zero claims is the consideration of near-miss events, which refer to incidents or situations where a potential collision or accident was narrowly avoided. For example, if a driver abruptly brakes or swerves to avoid hitting another vehicle, pedestrian, or obstacle, the telematics system can recognize this as a near-miss event. Stipancic et al. (2018) compared hard braking and accelerating events with historical crash data using Spearman's correlation and pairwise Kolmogorov-Smirnov (K-S) tests, finding a positive correlation between these factors and crash frequency. Guillen et al. (2020) used negative binomial regression models to predict near-miss events, including dangerous turning, hard braking, and hard acceleration. By incorporating near-miss events into risk assessment models, insurers can gain a deeper understanding of driver behavior and better evaluate accident risk. Furthermore, Guillen et al. (2021) proposed a model that incorporates both a basic premium and a penalization factor based on the occurrence of near misses such as hard-braking, hard-acceleration and use of smartphone. This approach incentivizes policyholders to adopt safer driving practices by directly linking their driving behavior to the cost of insurance coverage.

In summary, the incorporation of telematic data and the use of advanced machine learning techniques have significantly improved risk assessment and premium calculation in automobile insurance. The combination of traditional covariates with telematic data, along with the exploration of innovative approaches like zero-inflated regression models and near-miss event analysis, has provided insurers with more accurate tools for assessing risk and setting premiums based on individual driving behaviour.

The main aim of this thesis is to serve as a survey of models and methods for the use of telematic data in auto insurance rating. The models are illustrated through a comparative analysis of their performance on a synthetic publicly available dataset that was generated from the experience of a Canadian-based insurance company. Hopefully this survey and analysis can be useful to actuaries and actuarial students as an introduction to telematic data in ratemaking.

1.4 Structure

This study is structured as follows. Section 2 describes the dataset and explains the preprocessing of data to prepare the dataset for machine learning methods. Moreover, the variance inflation factors is evaluated to assess the multicollinearity the dataset used in this study. Section 3.1 establishes risk assessment models using both traditional and telematic risk factors, and it compares the prediction performances of four machine learning methods, namely logistic regression, random forests, gradient boosting trees, and feed-forward neural network. Moreover, two feature importance evaluation techniques are applied on machine learning techniques to assess the impact of each predictor. Section 3.4 concludes the research by presenting the findings.

2. *Data*

In motor insurance, actuaries traditionally determine premiums based on customer profiles and claims history. However, the advent of telematic data has revolutionized the industry by providing insurers with access to valuable driving behavior information. Telematic data encompasses factors such as speed, acceleration, time of driving, engine RPM, and distance driven, which are collected by a device installed in the vehicle while the driver is behind the wheel. These features serve as indicators of the insured individual's driving style. The development of telematic data led to usage-based insurance (UBI), enabling insurers to make more accurate risk assessments and determine premiums based on individualized measurements, as discussed in Arumugam and Bhargavi (2019). Therefore, a policyholder with a risky driving behavior is charged a higher premium.

There are several approaches to collect telematic data such as black boxes, dongles, embedded equipment, and smart-phones, as outlined in Arumugam and Bhargavi (2019). A black box refers to an electronic device installed in the vehicle that records accident-related information. It allows for one-way interaction, and the data is typically accessible only after an accident has occurred. Similarly, a dongle is an electronic device installed in the vehicle, enabling a server to access the vehicle network. It also functions as a one-way interaction device. Car manufacturers sometimes provide embedded equipment in the vehicles to record telematic data. Some examples of embedded equipment are the remote diagnosis device infotainment services and navigation sensors. Another method for collecting telematic data is through smart phones. Smart-phones can connect to a device or operate as stand-alone devices for data collection. The built-in sensors in smart-phones facilitate the acquisition of driving variables such as speed, hard braking, hard acceleration and hard cornering. This method is cost-efficient and entails less computational complexity. Additionally, the use of Global Positioning System (GPS) technology enables the collection of telematic data. GPS signals provide information such as speed, acceleration and braking that can be derived from the signals of GPS. This method provides accurate data. However, the cost of implementation and complexity of the calculations are high. The in-vehicle data recording devices can also provide additional services such as automatic emergency calls, stolen vehicle monitoring and economically and more convenient driving suggestions as highlighted in Baecke and Bocca (2017).

2.1 Data description

This section describes the dataset used in this study and some illustrations of the data are presented. The data used in this survey is a synthetic data available in So et al. (2021). The synthetic data is emulated from a real dataset acquired from a Canadian-based insurer, that offers a UBI program to its automobile insurance policyholders. The emulation of the data consist of a three-stage process. First, using an extended Synthetic Minority Oversampling Technique (SMOTE) algorithm, a synthetic portfolio of the space of feature variables is produced. The SMOTE algorithm produces new synthetic data points using the original data point using K nearest neighbors. In the second stage, the values of the number of claims are simulated using a feed-forward neural network. In the last step, the aggregated amount of claims are simulated using a feed-forward neural network with the number of claims included in the predictors. According to statistics and illustrations available in So et al. (2021), the synthetic data shows remarkably similar statistics to the real dataset. The synthetic dataset consist of 100,000 policies. Table 2.1 provides an overview of all variables in the dataset and included in the models.

Type	Variable Name	Type of Variable	Description
Traditional	Duration	Integer	Duration of the insurance coverage of a given policy and year, in days ranging between [27,366]
	Insured.age	Integer	Age of insured driver, in years ranging between [16,103]
	Car.age	Integer	Age of vehicle, in years ranging between [-2,20], negative values are possible as buying a newer model can be up to two years in advance
	Credit.score	Double	Credit score of insured driver ranging between [422,900]
	Annual.miles.drive	Double	Annual miles expected to be driven declared by driver ranging between [0, 56731.17]
	Years.noclaims	Integer	Number of years without any claims ranging between [0,79]
Telematic	Annual.pct.driven	Double	The number of days a policyholder uses vehicle divided by 365
	Total.miles.driven	Double	Average distance driven in miles per day during the observation [0,0.67274]
	Pct.drive.xxx	Double	Percent of driving day xxx of the week: mon/tue/. . . /sun. Note that these variables are compositional meaning that the sum of seven variables is 1.
	Pct.drive.xhrs	Double	Percent vehicle driven within x hrs: 2hrs/3hrs/4hrs
	Pct.drive.xxx	Double	Percent vehicle driven during xxx: wkday/wkend. Note that these variables are compositional
	Pct.drive.rushxx	Double	Percent of driving during xx rush hours: am/pm
	Avgdays.week	Double	Mean number of days used per week, ranging between [0,7]
	Accel.xxmiles	Double	Number of sudden acceleration 6/8/9/. . . /14 mph/s per 1000 miles, ranging between [0,621]
	Brake.xxmiles	Double	Number of sudden brakes 6/8/9/. . . /14 mph/s per 1000 miles, ranging between [0,621]
	Left.turn.intensityxx	Double	Number of left turns per 1000 miles with intensity 08/09/10/11/12, ranging between [0, 794740]
	Right.turn.intensityxx	Double	Number of right turns per 1000 miles with intensity 08/09/10/11/12, ranging between [0, 841210]
Response	NB_Claim	Integer	Number of claims during observation, ranging between [0,3]
	AMT_Claim	Double	Aggregated amount of claims during observation, ranging between [0, 550.66]

Table 2.1: Description of synthetic dataset variables.

The synthetic dataset contains 47 variables, which can be categorized into three groups: (1) 6 traditional variables such as age of driver, number of years without claim and age of car, (2) 39 telematic variables such as total miles driven, number of brakes, and number of hard accelerations, and (3) two response variables describing the number of claims and the amount of claims. Note that 95.72% of observations have zero claims, 4.06% have one claim,

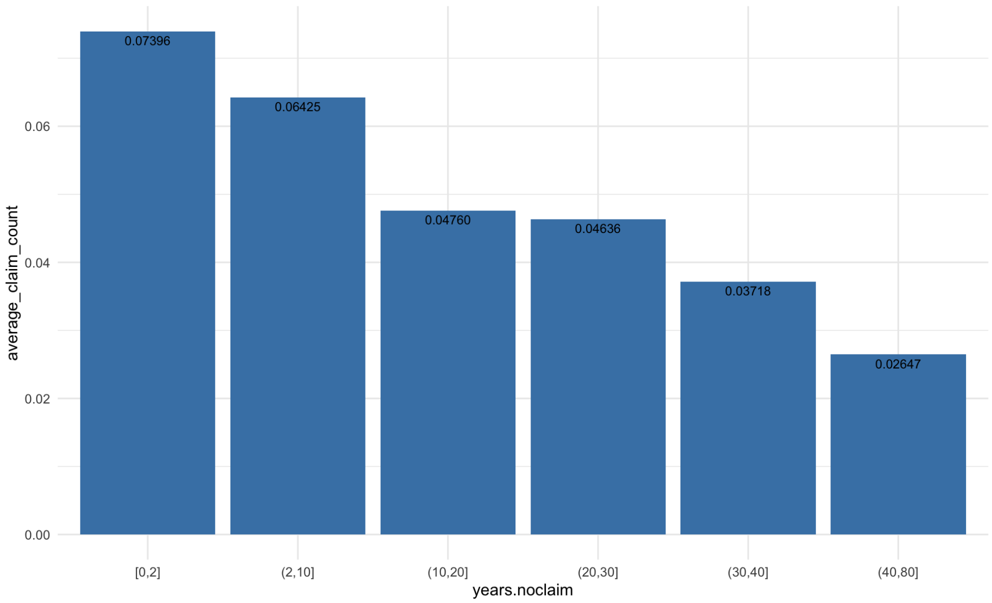


Figure 2.1: The relationship between the average number of the drivers with claims and drivers without any claims during the last $0-2$ years, $2-10$ years, $10-20$ years, $20-30$ years, $30-40$ years and more than 40 years.

0.20% have two claims, and 0.007% have three claims.

The traditional and telematic variables can be used as predictors in the machine learning methods to predict the response variable. Visualization of the data set can help detecting possible relationships between the variables. For example, Figure 2.1 illustrates the relationship between the number of the years without any claims and the average number of high-risk drivers(drivers with at least one claim during the observation). The figure shows that the drivers with more years without claims are safer drivers. Figure 2.2 demonstrate the statistics for rush hours driving. According to this figure the risk of accident is higher for drivers that use their car during rush hours specially at nighttime. Figure 2.3 shows the average count of left and right turns for high-risk drivers and low-risk drivers. According to this figure, the risk of having accident is high when drivers are turning, specially for left turns. There are more data illustrations available in Appendix 3.4.

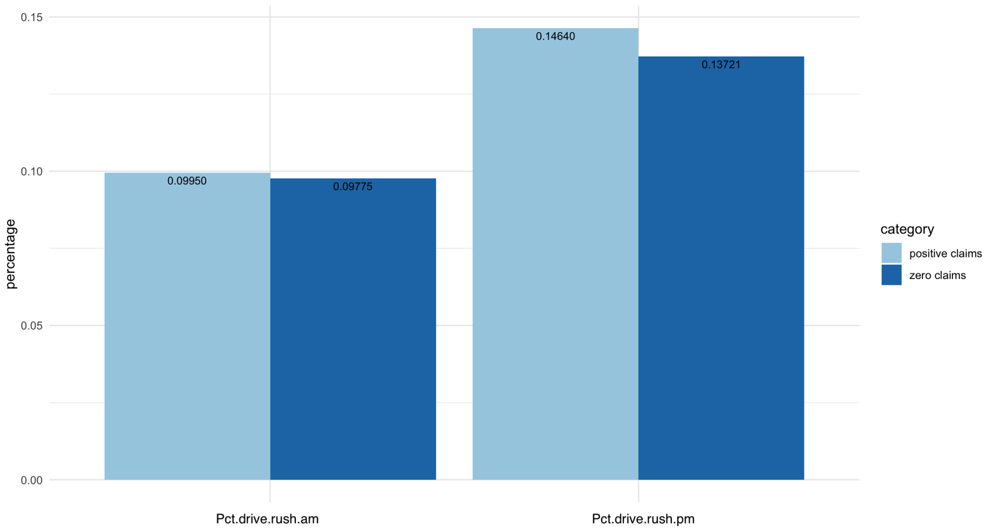


Figure 2.2: The average percentage of driving during am rush hours and pm rush hours for drivers with at least one claim and drivers without any claim.

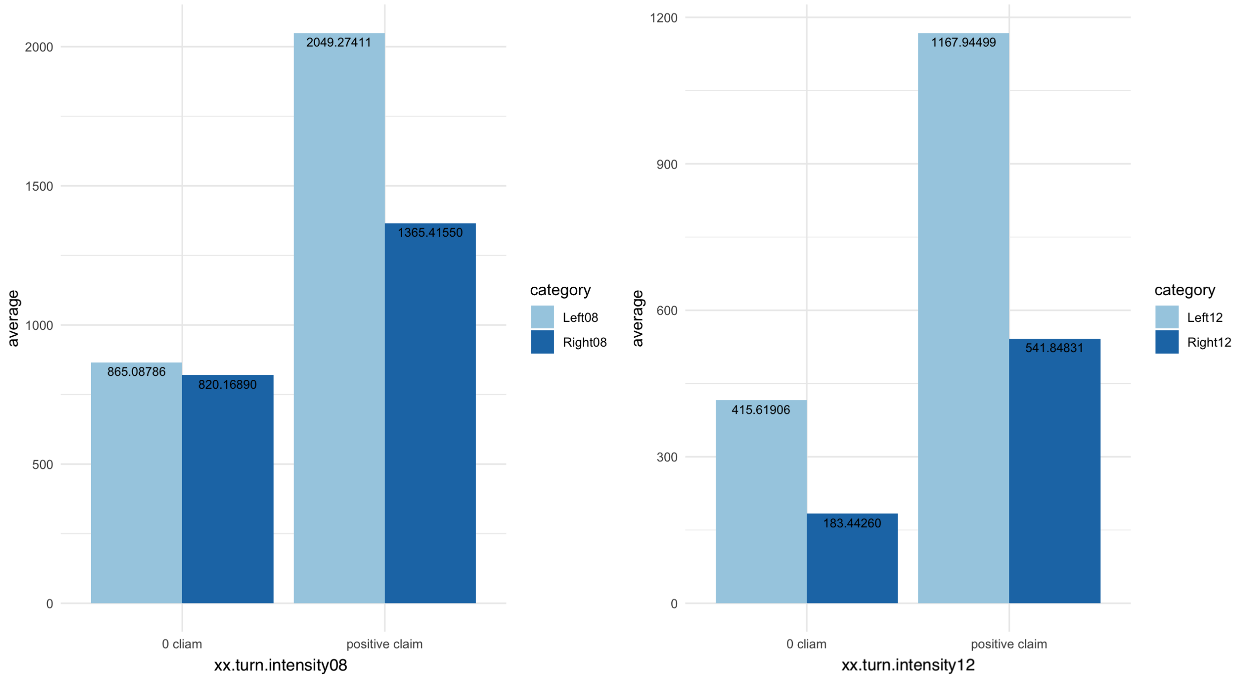


Figure 2.3: The average count of left turns and right turns for drivers with at least one claim and drivers without any claim. The left figure is for the turns with intensity equal to 8 and the right figure is for the turns with intensity equal to 12.

2.2 Multicollinearity

The data used in this study exhibits multicollinearity. When the correlation between the variables is high, it undermines the statistical significance of an explanatory variable. Moreover, the coefficient becomes very sensitive to small changes in the model.

Using the variance inflation factor (VIF) one can identify the multicollinearity in regression analysis. Multicollinearity happens when there exists correlation between independent variables in a model. VIFs are usually calculated numerically as part of a regression analysis. In particular, the variance inflation factor for the j^{th} predictor is given by:

$$VIF_j = \frac{1}{1 - R_j^2},$$

where R_j^2 is the R^2-value obtained by regressing the j^{th} predictor on the remaining predictors. VIF values vary between 1 to positive infinity, and a VIF equal to 1 shows there is no correlation between the variables. If a VIF is greater than 10 then the multicollinearity is high. One can eliminate the effect of the multicollinearity by removing the highly correlated variables from the model.

The dataset used in this study exhibits multicollinearity, which causes inaccurate predictions and over-fitting in machine learning methods. The severity of the multicollinearity is measured here using the Variance Inflation Factor (VIF). Table 2.2 illustrates VIF values of the variables in the dataset. The predictors with VIF value higher than 10 are considered highly correlated. To eliminate the effect of multicollinearity, a subset of predictors are considered in the models, which are mentioned in Table 2.3 together with their VIF values. Note that in the new subset of variables, all variables with VIF more than 10 were eliminated, so the VIF values are less than 10, except for “Accel.14miles” and “Brake.14miles”. These two variables are included in the model since hard acceleration and brake have high impact on the risk of accident. Therefore, the variables used in this study to assess the risk of accident are all listed in Table 2.3

Variable	VIF value
Duration	0.6008138
Insured.age	2.94749
Car.age	3.071365
Credit.score	1.280033
Annual.miles.drive	0.6033964
Years.noclaims	2.812487
Annual.pct.driven	1.011482
Total.miles.driven	0.2054459
Pct.drive.2hrs	1.228046
Pct.drive.3hrs	0.4165208
Pct.drive.4hrs	0.3808286
Pct.drive.wkend	1.523781×10^{15}
Pct.drive.wkday	4.091993×10^{15}
Pct.drive.rush.am	0.3133676
Pct.drive.rush.pm	0.7547143
Avgdays.week	0.1732634
Pct.drive.mon	1.302230×10^{15}
Pct.drive.tue	7.350217×10^{15}
Pct.drive.wed	6.695851×10^{14}
Pct.drive.thr	9.314201×10^{14}
Pct.drive.fri	2.49742×10^{14}
Pct.drive.sat	9.359063×10^{14}
Pct.drive.sun	9.438624×10^{14}

Variable	VIF value
Accel.06miles	2.293262
Accel.08miles	14.28145
Accel.09miles	41.27826
Accel.11miles	114.4122
Accel.12miles	149.9056
Accel.14miles	138.1172
Brake.06miles	9.167488
Brake.08miles	24.75065
Brake.09miles	60.06385
Brake.11miles	87.29104
Brake.12miles	114.9023
Brake.14miles	54.91514
Left.turn.intensity08	116.5550
Left.turn.intensity09	631.2543
Left.turn.intensity10	1294.965
Left.turn.intensity11	6018.608
Left.turn.intensity12	1658.123
Right.turn.intensity08	97.18742
Right.turn.intensity09	597.4358
Right.turn.intensity10	2908.332
Right.turn.intensity11	6552.246
Right.turn.intensity12	904.1544

Table 2.2: Variance Inflation Factor (VIF) values of traditional and telematic predictors in logistic regression. The predictors with VIF values higher than 10 are considered highly correlated.

Variable	VIF value
Duration	1.586643
Insured.age	3.176990
Car.age	1.072250
Credit.score	1.218741
Annual.miles.drive	1.205720
Years.noclaims	3.069423
Annual.pct.driven	2.141902
Total.miles.driven	2.389323
Pct.drive.2hrs	2.411448
Pct.drive.3hrs	3.800461
Pct.drive.4hrs	2.324903

Variable	VIF value
Pct.drive.wkend	1.087751
Pct.drive.rush.am	1.279645
Pct.drive.rush.pm	1.274341
Avgdays.week	1.359456
Accel.06miles	1.793109
Accel.14miles	25.571923
Brake.06miles	1.798006
Brake.14miles	25.318923
Left.turn.intensity08	3.873734
Left.turn.intensity12	3.878388
Right.turn.intensity08	4.311204
Right.turn.intensity12	4.317575

Table 2.3: Variance Inflation Factor (VIF) values of traditional and telematic predictors in logistic regression. The predictors with VIF values higher than 10 are considered highly correlated.

2.3 Imbalanced data

The response variable in the models included in this study is a binary variable, which takes a value of 1 if the policyholder makes at least one claim during the observation, and zero if the policyholder makes zero claims during the observation. As a result, this response variable is equal to 0 for 95.728% of the policyholders in this study and 1 for only 4.272% of the observations. Therefore, the dataset is imbalanced. Imbalanced data can pose challenges in machine learning classification techniques because algorithms tend to be biased towards the majority class; they can achieve high accuracy by simply predicting the majority class for most or all instances. However, this approach fails to capture the patterns and characteristics of the minority class.

When dealing with imbalanced datasets, accuracy is not a reliable performance metric since it can be misleading. Instead metrics such as the area under the receiver operating characteristic (ROC) curve (AUC) are more appropriate for evaluating model performance.

Dealing with imbalanced data requires careful consideration and appropriate techniques to ensure that machine learning models can effectively learn from all classes and make accurate predictions for both the majority and minority classes. Resampling techniques can be applied to rebalance the data (Fernández et al., 2018). Resampling techniques can be classified into three broad groups: (1) oversampling methods, (2) undersampling methods, and (3) hybrid methods. Oversampling involves creating a superset of the original dataset by replicating some instances or creating new instances from existing ones, while undersampling methods create a subset from the original dataset by eliminating instances (usually majority class instances). Hybrid methods aim to balance the distribution of the dataset by combining both sampling methods.

In a simple test we now assess the relevance of resampling methods for our data; three resampling techniques, including random oversampling, random undersampling, and hybrid methods, are applied to rebalance the dataset. Random oversampling involves randomly selecting observations from the minority class, with replacement, to create the training set. Random undersampling technique randomly selects examples from the majority class, and omits them from training set. Note that resampling techniques are only applied to the training dataset, and are not applied to the test set, which is for evaluation of the model performance. Oversampling may increase the likelihood of overfitting the minority class, and undersampling results in a loss of data, which might make the decision boundary between the two classes harder to learn. Interesting results are usually achieved by combining both random oversampling and undersampling. A modest amount of oversampling of the minority class can improve the bias towards this class, and a modest amount of undersampling of the majority class reduces the bias on the majority class. Table 2.4 presents the AUC values for the logistic regression, using both traditional and telematic variables. Four different training datasets are used in the models, and according to the results of Table 2.4, resampling methods are not improving the model performance. Therefore, in this study, the original data, without resampling, is used with machine learning techniques.

	raw data	undersampling	oversampling	combination
AUC-training set	0.7836497	0.7891712	0.7838904	0.7856054
AUC-test set	0.7844003	0.7854597	0.7856575	0.785725

Table 2.4: Four training datasets are analyzed using logistic regression. The AUC-ROC test is used for model evaluation, and the results are similar for all training sets.

3. *Methodology*

3.1 Predictive models

The impact of telematic data on assessing the likelihood of an accident can be effectively explored through the application of machine learning techniques. Regulators in insurance companies often insist on using “white box” learning methods which are easy to interpret. Examples of such methods include GLMs (Generalized Linear Models) and logistic regression. Unfortunately, these methods are limited in their ability to identify complex non-linear relationships. On the contrary, “black box” learning methods like random forests and neural networks can detect non-linear relationships with greater accuracy. In this section, the probability of having a claim for a policyholder according to the traditional and telematic information is investigated using different machine learning methods. Four machine learning methods are used for the analysis, and their performances are compared using statistical metrics.

The four machine learning methods considered in this study are logistic regression, random forests, gradient boosting trees, and feed-forward neural networks. The logistic regression is one of the common methods in insurance industry for binary classification because it is easy to interpret. However, it is limited in its ability to capture complex non-linear relationships. To address this limitation, more sophisticated methods such as random forests, gradient boosting trees, and neural networks were also included in the study to uncover the underlying complex relationships within the data. Random forests, an ensemble classifier composed of multiple decision trees, improve upon the limitations of individual decision trees by aggregating the outcomes of multiple trees. By mitigating over-fitting, random forests generally outperform decision trees alone. The gradient boosting trees, another ensemble classifier, produce a prediction model in the form of an ensemble of weak decision trees. This method usually outperforms random forests. Artificial neural networks are a powerful method known for their capacity to learn and model non-linear and complex relationships. They have demonstrated strong performance across a wide range of applications, making them an important consideration for this study.

The rest of this section offers a detailed and comprehensive explanation of each of the aforementioned methods, delving into their underlying principles and intricacies. Additionally, a brief description of the implementation of these methods is presented to provide a practical understanding of how they are applied in the context of this study.

3.1.1 Logistic regression

Logistic regression is a supervised learning method which is used for classification. This method is a generalized linear method with a specific link function to model the relationship between the predictors and the response variable. In the following the logistic regression for binary classification is explained. Assume all observations are independent, and given the predictor X_i, the response value Y_i for $i = 1, \ldots, n$ has a Bernoulli distribution. The logistic function denotes as ζ is used as the inverse link function:

$$\mathbb{P}[Y_i = 1|X_i] = \mathbb{E}[Y_i|X_i] = \zeta(\tilde{X}_i^T \beta) := \frac{\exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j})}{1 + \exp(\beta_0 + \sum_{j=1}^{p} \beta_j X_{i,j})}$$

where $\tilde{X}_i^T = \begin{bmatrix} 1 & X_{i,1} & \ldots & X_{i,p} \end{bmatrix}$, and the logistic function, also known as the sigmoid function, maps the linear combination of predictors to a probability value between 0 and 1. By estimating the model parameters using maximum likelihood estimation, logistic regression determines the optimal decision boundary that separates the two classes based on the given predictor variables. The likelihood functions is given by

$$\mathcal{L}(y; \theta, \phi) = \sum_{i=1}^{n} \log f(y_i; \theta, \phi)$$

according to the assumptions of the logistic regression, given X_i, the Y_i are all distributed from the same class of exponential family of distributions. Indeed, given X_i the pdf of Y_i is assumed to be

$$f(y_i; \theta, \phi) = \exp\Big(\frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\Big)$$

for some functions b, c, a_i and θ_i for $i = 1 \ldots, n$. Therefore, the log-likelihood function is equal to

$$\begin{aligned} \mathcal{L}(y; \theta, \phi) &= \sum_{i=1}^{n} \log f(y_i; \theta, \phi) \\ &= \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi). \end{aligned}$$

The likelihood is usually impossible to optimize in closed-form, and it needs to be maximized numerically. Then, a new observation with predictors X_0 is classified according to

$$\begin{aligned} Y_0 &= 1 \qquad if \quad \hat{\mathbb{P}}[Y_0 = 1|X_0] \geq c, \\ Y_0 &= 0 \qquad \text{otherwise} \end{aligned}$$

where $\hat{\mathbb{P}}[Y_0 = 1|X_0] = \zeta\Big(\hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j X_{0,j}\Big)$ and c is the cut-point, see James et al. (2013) for more details. This method is implemented to the dataset by using '$glm()$' function in R.

3.1.1.1 Lasso regularization

One approach to reduce the variance of prediction and enhance the prediction accuracy and interpretability is using regularization methods. Regularization methods shrink the parameters toward zero by applying specific constraints on them. The lasso regression is a regression analysis method which performs regularization and variable selection. By considering negative log-likelihood as the loss function, $L(y_i, p_i) = -\{y_i \log \hat{p_i} + (1 - y_i)\log(1 - \hat{p_i})\}$, the lasso coefficients $\hat{\beta}_{\lambda}^{lasso}$ minimize the following quantity

$$\hat{\beta}_{\lambda}^{lasso} = \arg\min_{\beta} \sum_{i=1}^{n} L(y_i, p_i) + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Note that p_i depends on β. Moreover, $\lambda > 0$, which can be selected using cross-validation by optimizing out-of-sample performance. Note that in lasso regression, the penalty term is $\lambda \sum_{j=1}^{p} |\beta_j|$, which prevents the parameters from moving freely. The lasso regression is usually impossible to solve in closed-form. However, by using numerical algorithms, one can estimate the lasso coefficients. In the *R* software, one can use *glmnet*() function to do the lasso regression.

3.1.2 Decision trees

Tree-based methods are simple and interpretable machine learning techniques that can be applied to both regression and classification problems. This methods involve segmenting the predictor space into a number of simple regions, and predicting the outcome of a new observation based on the mode (for classification problems) or the mean (for regression problems) of the training observations within the corresponding region. The decision tree method consists of two fundamental steps. First it partitions the predictor space into J distinct and non-overlapping regions denoted as $R_1, \ldots, R_J$. Then, for every observation falling into region R_j, the method assigns the same prediction, typically the mean or the mode of the observations within that specific region, see James et al. (2013) for more details.

Finding an appropriate partition for the predictor space is an important step in decision trees. It is computationally impossible to consider every possible partition of the predictor space. One approach to find a proper partition is by using a technique called "recursive binary splitting". This technique follows a top-down and greedy strategy. It starts at the top of the tree and iteratively splits the predictor space based on the best split at each step. The splitting process is continues until a predefined stopping criterion is met, such as when all regions contain fewer than five observations. It is worth noting that, in classification problems, the classification error is not a sensitive metric for training the model. Instead, metrics like the Gini index are typically used as a more suitable loss function. The Gini index associated with a subregion r quantifies the total variance across all classes within the subregion r and can be defined as follows:

$$G_r = \sum_{g \in \mathcal{G}} \hat{p}_{rg}(1 - \hat{p}_{rg}),$$

where $\hat{p}_{rg}$ is the proportion of observations with response g among all observations whose predictors fall within region r, see James et al. (2013). It is worth noting that the Gini-index, denoted as G_r, takes small values if all $\hat{p}_{rg}$ are close to either zero or one. Thus, the Gini index serves as a measure of node purity, with small values indicating that a node mostly consists of observations from a single class. Node purity provides a higher certainty in a predictions. Therefore, the recursive binary splitting performs as following for binary classification decision trees:

For any given j and s, define:

$$R_1(j, s) = \big\{X | X_j < s\big\} \quad and \quad R_2(j, s) = \big\{X | X_j \geq s\big\},$$

where the value of j and s can be determined by minimizing the weighted average of the Gini index of the two subregions:

$$\pi_1 G_{R_1} + \pi_2 G_{R_2} = (\frac{n_{r_1}}{n_{r_1} + n_{r_2}}) \sum_{g \in \mathcal{G}} \big[\hat{p}_{R_1 g}(1 - \hat{p}_{R_1 g})\big] + (\frac{n_{r_2}}{n_{r_1} + n_{r_2}}) \sum_{g \in \mathcal{G}} \big[\hat{p}_{R_2 g}(1 - \hat{p}_{R_2 g})\big],$$

where $\hat{p}_{R_i g}$, for $i = 1, 2$, represents the proportion of observations with response g among all observations whose predictors fall within region R_i, and n_{r_j}, for $j = 1, 2$ is the number of observations in r_i (James et al., 2013).

Although decision trees are equipped with a stopping criterion, overfitting remains a potential issue. To address this, constructing smaller trees with fewer splits can effectively mitigate variance and improve interpretability. Moreover, the simplicity of single decision trees enables them to closely resemble human decision-making patterns, facilitating easy interpretability. However, it is worth noting that their predictive accuracy tends to be lower when compared to alternative learning methods. One approach to improve the predictive performance of decision trees is aggregating multiple decision trees. Techniques such as bagging, random forests, and boosting involve combining multiple trees to collectively produce more accurate predictions.

3.1.2.1 Random forests

Random forests is a powerful technique that combines decision trees to enhance the predictive performance. The process of implementing random forests involves the following steps:

1. Creating B distinct training sets.

2. Building a decision model for each training set. The prediction function for the model b is denoted as $\hat{f}_b$. Note that during the process of growing the tree, the splitting is performed only on m random predictors (often m is chosen as the square root of the total number of predictors, i.e., $m = \sqrt{p}$) at each splitting stage.

3. The final prediction for an observation is obtained by averaging the predictions $\hat{f}_{avg}(X) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}_b(X)$.

Splitting the training set into B subsets, to create B different training sets, might generate small subsets. Therefore, the technique of "bootstrapping" can be used to generate B different larger training sets. This approach creates each training set by sampling n times with replacement from the original training set. In the context of this study, the random forests method is applied using the $randomForest()$ function in R, which implements Breiman's random forest algorithm (Breiman, 2001) for classification and regression. This algorithm uses bootstrapping to generate B distinct training sets. This ensures that each training set is sufficiently large and diverse, enhancing the effectiveness of the random forests algorithm.

If the splitting is performed on every predictor without any constraint, there is a risk of all trees having splits on the same strong predictor early on, resulting in similar trees. This high similarity in trees limits the potential improvement in prediction. In Breiman's random forests algorithm, a technique to intriduce dissimilarity among the trees is employed to avoid the issue of generating highly correlated trees. In this technique, random forests introduce randomness by selecting a subset of predictors at each splitting stage. Typically, a random number m (often set as $m = \sqrt{p}$) predictors is chosen, and the split is performed only on this subset of predictors. This procedure ensures that the random forests algorithm creates diverse decision trees, leading to predictions with reduced correlation.

In this study, the random forest method is implemented using the $randomForest$ package and the $ISLR2$ package. These packages provide the necessary functionalities for constructing and analyzing random forests.

3.1.3 Gradient boosting trees

Boosting is versatile method that can be applied on regression and classification problems. It involves constructing an ensemble of weak prediction models, such as decision trees, smoothing splines, or neural networks. Among these weak learners, decision trees are widely accepted and commonly used in ensemble techniques. In gradient boosting trees, the algorithm iteratively builds decision trees, where each subsequent tree corrects the prediction errors made by the previous trees by minimizing the loss function using gradient descent.

Ensemble techniques have the following structural form:

$$g(\mathbf{E}[Y|X=x]) = F_M(x) = \sum_{m=1}^{M} \beta T(x; \boldsymbol{a}_m),$$

where g is the link function (where in classification problems, g is logit function), β is a fixed learning rate, which can be determined using the validation set (in this study the value is set to 0.05), and $T(x; \boldsymbol{a}_m)$, for $m = 1, \ldots, M$ are simple functions of the features x, and characterized by parameters $\boldsymbol{a}_m$, see Denuit and Trufin (2019) for more details. In boosting trees, the function $T(\cdot; \boldsymbol{a}_m)$ is a decision tree, where $\boldsymbol{a}_m$ represent the splitting variables and their split values as well as the corresponding predictors in the terminal nodes. Note that using decision trees as a weak learner requires a stopping criterion, such as determining maximum interaction depth, to avoid overfitting. Now for observations $\{(x_i, y_i)\}_{i=1}^n$, one can estimate the coefficients and parameters by minimizing the following equation:

$$\min_{\boldsymbol{a}_m} \sum_{i=1}^{n} L\Big(y_i, g^{-1}(F_M(x))\Big) = \min_{\boldsymbol{a}_m} \sum_{i=1}^{n} L\Big(y_i, g^{-1}\big(\sum_{m=1}^{M} \beta T(x_i; \boldsymbol{a}_m)\big)\Big).$$

Let $\hat{z}_i := F_m(x_i) = F_{m-1}(x_i) + \beta T(x_i; \boldsymbol{a}_m)$ for $i = 1, \ldots, n$. A proper activation function in binary classification problems is the sigmoid function. Therefore, $\hat{p}_i = Pr[y_i = 1|x_i] = \frac{1}{1+e^{-\hat{z}_i}}$, see Denuit and Trufin (2019) for more details. Moreover, suitable loss function is the binary cross-entropy loss, and the formula for binary cross-entropy loss is as follows (Denuit and Trufin, 2019):

$$L_i = -y_i \log \hat{p}_i - (1 - y_i) \log(1 - \hat{p}_i).$$

Now, one can estimate the parameters by minimizing the loss function as below

$$\begin{aligned}
\hat{\boldsymbol{a}}_m &= \arg \underset{\boldsymbol{a}_m}{min} \sum_{i=1}^{n} L_i(y_i; g^{-1}F_m(x_i)) = \arg \underset{\boldsymbol{a}_m}{min} \sum_{i=1}^{n} L_i(y_i; \hat{p}_i) \\
&= \arg \min_{\boldsymbol{a}_m} \sum_{i=1}^{n} \Big[- y_i \log \hat{p}_i - (1 - y_i) \log(1 - \hat{p}_i) \Big] \\
&= \arg \min_{\boldsymbol{a}_m} \sum_{i=1}^{n} \Big[- y_i \log \frac{\hat{p}_i}{1 - \hat{p}_i} - \log(1 - \hat{p}_i) \Big].
\end{aligned}$$

Knowing that $\hat{y}_i = \log \frac{\hat{p}_i}{1-\hat{p}_i}$, one can get

$$\hat{\boldsymbol{a}}_m = \arg \min_{\boldsymbol{a}_m} \sum_{i=1}^{n} \Big[- y_i \hat{y}_i + \log(1 + e^{\hat{z}_i}) \Big]$$

Finding the solution to the optimization problem described above involves complex and time-consuming calculations. To address this, one can employ numerical optimization methods.

3.1.3.1 Steepest descent

The primary objective is to minimize the training sample of the loss function:

$$L(F_M(x)) = \sum_{i=1}^{n} L(y_i, g^{-1}(F_M(x_i)))$$

with respect to $F_M(x)$. Now, by ignoring the constraint that $F_M(x)$ is the sum of trees, one can view the optimization problem as the following numerical optimization:

$$\hat{\boldsymbol{\eta}} = \arg \min_{\boldsymbol{\eta}} L(\boldsymbol{\eta}),$$

where $\boldsymbol{\eta} \in \mathcal{R}^n$ are the values of this approximating function $g^{-1}(F(x_i))$ for the training observation point x_i:

$$\boldsymbol{\eta} = \{\eta_1, \ldots, \eta_n)' = (g^{-1}(F_M(x_1)), \ldots, g^{-1}(F_M(x_n))\}^T.$$

Interaction Depth	training set AUC	validation set AUC
1	0.8404	0.8007
2	0.8386	0.8325
3	0.9967	0.8537
4	0.9999	0.8644

Table 3.1: The AUC values of a gradient boosting trees with different interaction depths. All the parameters in the model are fixed and only the maximum depth of each tree in the model changes.

Numerical optimization methods often express the solution as

$$\hat{\boldsymbol{\eta}} = \sum_{t=0}^{T} b_T, \qquad b_t \in \mathbb{R}^n,$$

where b_0 is an initial guess and $b_1, \ldots, b_T$ are successive increments, each based on the preceding steps. In steepest descent numerical optimization, step b_t is defined as $b_t = \rho_t \theta_t$, where ρ_t is a scalar, and

$$\theta_t = \left(\left[\frac{\partial L(y_1, g^{-1}(\eta_1))}{\partial \eta_1} \right]_{\eta_1 = \hat{\eta}_{t-1,1}}, \ldots, \left[\frac{\partial L(y_n, g^{-1}(\eta_n))}{\partial \eta_n} \right]_{\eta_n = \hat{\eta}_{t-1,n}} \right)'$$

is the gradient of $L(\boldsymbol{\eta})$ evaluated at $\hat{\boldsymbol{\eta}}_{\boldsymbol{t-1}} = (\hat{\eta}_{t-1,1}, \ldots, \hat{\eta}_{t-1,n})'$ given by

$$\hat{\boldsymbol{\eta}}_{\boldsymbol{t-1}} = b_0 + b_1 + \cdots + b_{t-1}.$$

Note that the negative gradient θ_t gives the local direction along which $L(\boldsymbol{\eta})$ decreases the most rapidly at $\boldsymbol{\eta} = \hat{\boldsymbol{\eta}}_{\boldsymbol{t-1}}$. Note that the step length ρ_t is then determined by

$$\rho_t = \arg\min_{\rho > 0} L(\hat{\boldsymbol{\eta}}_{\boldsymbol{t-1}} - \rho \theta_t),$$

which provides the update $\hat{\boldsymbol{\eta}}_{\boldsymbol{t}} = \hat{\boldsymbol{\eta}}_{\boldsymbol{t-1}} - \rho_t \theta_t$.

In this study, the gradient boosting trees method is applied on the dataset using the '*xgboost*' package.

Boosting trees involve two crucial tuning parameters: the size of the trees and the number of trees, denoted as M. The size of the trees defines the complexity of the trees in the model. There exists different approaches to determine the complexity of a tree, such as specifying

Number of Trees	training set AUC	validation set AUC
1000	0.9526	0.8325
750	0.9383	0.8298
500	0.9177	0.8256
250	0.8824	0.8176

Table 3.2: The AUC values of gradient boosting trees with different number of trees (M). All the parameters in the model are fixed and only the number of trees in the model changes.

the maximum number of terminal nodes (leaves) or determining the maximum depth of each tree (the interaction depth). In '*xgboost*()' function, the complexity of a tree is determined by the interaction depth. Table 3.1 illustrates the AUC results for different interaction depth values on the validation set and the training set. As shown, increasing the interaction depth leads to higher AUC values for both sets. However, it increases overifitting significantly. Therefore, to prevent complexity and over-fitting, it is advisable to select smaller interaction depth values.

Another important parameter in the gradient boosting trees is the number of trees (M). Table 3.2 illustrates the AUC results for different number of trees. According to Table 3.2 when the number of trees increases, the AUC result also increases for both validation set and training set.

3.1.4 Feed-Forward neural networks

An artificial neural network is a machine learning technique that draws inspiration from the biological neural networks found in animals. A neural network relies on combining non-linear transformations of predictors to make predictions. A specific type of artificial neural network is called Feed-Forward Neural Networks, which can be applied on both quantitative and classification problems.

Single-layer Neural Network:

The parametric form of a feed-forward neural network is briefly described below. Consider the predictors $X = (X_1, X_2, \ldots, X_p)$ and the response variable Y (which can be multivariate).

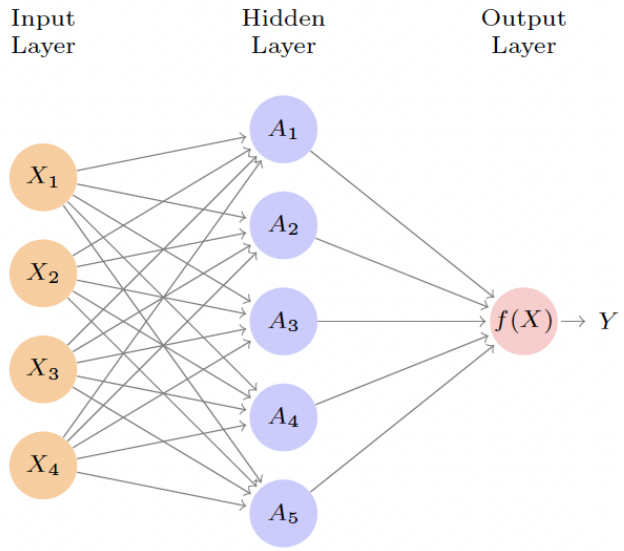


Figure 3.1: The structure of a single layer feed-forward neural network. In this model there exist four predictors X_1, X_2, X_3 and X_4. The hidden layer computes the activations, which are denoted by $A_k = h_k(X)$, for $k = 1, \ldots, K$. The activation functions, $h_k(.)$, are not observed and they need to be trained. The output layer is a linear combination of the activation functions such that $f(X) = \beta_0 + \sum_{k=1}^{5} \beta_k A_k$. Source: from James et al. (2013).

Neural networks construct a nonlinear function $f(X)$ to predict the response variable. Figure 3.1 depicts the structure of a single-layer neural network. A single-layer neural network consists of three layers: input layer, hidden layer and output layer. The input layer contains p units, with $X_1, \ldots, X_p$ representing the units. The hidden layer consist of an arbitrary number of nodes, which can be determined using the validation set. The hidden layer calculates the activation functions (nodes) $A_k = h_k(X)$, for $k = 1, \ldots, K$. These activation functions are obtained by applying nonlinear transformations to linear combinations of the inputs $X_1, \ldots, X_p$. Hence, the activation functions in the hidden layer are computed as follows:

$$A_k = h_k(X) = h_k(w_{k,0} + \sum_{j=1}^{p} w_{k,j} X_j),$$

where $h_k(.)$ is a nonlinear activation function, which is specified in advance. There exists various options for activation functions including sigmoid and ReLU. The sigmoid activation function is also used in logistic regression to transform the linear function to probabilities. It can be represented as follows:

$$h(z) = \frac{e^z}{1 + e^z}.$$

Rectified Linear Unit (ReLU) is contemporary choice of activation function which is characterized as follows:

$$h(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise.} \end{cases}$$

ReLU activation widely adopted due to its computational efficiency and memory storage benefits. Once these K activations are obtained, they are fed into the output layer. Now, denote $Z_m = \beta_{m,0} + \sum_{k=1}^{K} \beta_{m,k} h_k(X)$. Then for classification problems function $g(.)$ is a softmax function for classification problems with m classes, and the output is calculated as bellow:

$$\begin{aligned} S_m(X) = P[Y = m|X] &= g(\beta_{m,0} + \sum_{k=1}^{K} \beta_{m,k} h_k(X)) \\ &= g_m(Z) = \frac{e^{Z_m}}{\sum_{k=1}^{M} e^{Z_k}} \end{aligned}$$

where the parameters $\beta_0, \ldots, \beta_K$ and $w_{1,0}, \ldots, w_{K,p}$ need to be estimated. Therefore, $S_m(X)$ is the predicted probability that the observation X belongs to class m, $m = 1, \ldots, M$, see Bishop and Nasrabadi (2006) for more detail.

Fitting a Neural Network:

Training a neural network model and estimating the parameters involve minimizing the loss function across all observations. For binary classification problems, a suitable loss function is cross-entropy, which can be defined as follows:

$$E = -(y_i \log p_i + (1 - y_i) \log(1 - p_i)),$$

where p_i is the probability that an obsevation belongs to class $i = 0, 1$. Therefore,

$$\min_{\boldsymbol{\beta},\boldsymbol{w}} \sum_{i=1}^{n} E(y_i, f(x_i)) = \min_{\boldsymbol{\beta},\boldsymbol{w}} \sum_{i=1}^{n} E(y_i, \hat{p}_i) = \min_{\boldsymbol{\beta},\boldsymbol{w}} \sum_{i=1}^{n} \Big[-(y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)) \Big]. \tag{3.1}$$

However, solving this problem becomes challenging due to its non-convex parameter space, leading to the possibility of multiple solutions. Additionally, the complexity of this problem often makes it difficult and time-consuming to find a suitable solution. One approach to addressing this is through gradient descent. In this method, let θ denote the vector of parameters to be estimated, and the objective function is defined as shown in Equation 3.1. The gradient descent algorithm operates as follows:

1. Start with a random θ^0 and set $t = 0$.
2. Find a vector δ such that $\theta^{t+1} = \theta^t + \delta$ reduces the objective function, and set $t \leftarrow t+1$.
3. go to Step 2 if $E(\theta^{t+1}) < E(\theta^t)$, otherwise stop.

It should be noted that gradient descent can only guarantee the discovery of a local minimum, and the choice of the initial random vector θ^0 may lead to different local optima. To determine the direction δ for updating θ, one can compute the partial derivatives of $E(\theta)$ and evaluate them at the current value of $\theta = \theta^m$:

$$\nabla E(\theta^m) = \frac{\partial E(\theta)}{\partial \theta}\Big|_{\theta=\theta^m},$$

nodes count (size)	training set AUC	validation set AUC
2	0.8004018	0.7874062
5	0.8128508	0.797435
8	0.8196857	0.79894
10	0.8244184	0.8010579
15	0.8344605	0.8011259
30	0.8601271	0.804913

Table 3.3: The AUC values of a single-layer feed-forward neural network with different number of nodes. All the parameters in the model are fixed and only the number of nodes in the hidden layer changes. The AUC results on the validation dataset shows that 10 nodes in the hidden layer results in best performance.

which gives the direction in which $R(\theta)$ increases the most. Therefore, one can define the update direction as $\delta = -\rho \nabla E(\theta^m)$, where ρ represents a small learning rate. When the gradient vector becomes zero, it indicates that $E(\theta)$ has reached a minimum. Note that calculating the derivatives is simple using the chain rules. This procedure is known as "backpropagation" in the neural network literature. In fact, backpropagation efficiently calculates the gradients by computing the gradients one layer at a time and it iterates backward to prevent redundant calculations.

The gradient descent method often requires a considerable number of iterations to reach a local minimum, which can be time-consuming. One approach to accelerate the process is using stochastic gradient descent instead. Instead of computing derivatives for all n observations to find a local minimum, SGD samples only a small fraction of the observations known as a "minibatch". The size of the minibatch can be determined using the validation set.

The neural network method used in this study is a single-layer feed-forward neural network, implemented using the *nnet* package in *R*. The hidden layer consists of 10 nodes which are determined through validation set analysis. Table 3.3 presents the AUC values corresponding to different numbers of nodes. In the process of determining the number of nodes, all other model parameters remain fixed, and only the number of nodes varies. To ensure

max iteration	training set AUC	validation set AUC
100	0.808669	0.7980243
200	0.8173433	0.8004857
400	0.8211446	0.8024053
600	0.8211735	0.8024967
1000	0.8211767	0.80249348

Table 3.4: The AUC values of a single-layer feed-forward neural network with different maximum iteration values for stopping criteria in SGD. All the parameters in the model are fixed and only the stopping criteria varies. The results shows that after 610 iterations, the algorithm converges and reach to a local optimum.

consistent results, the initial weights for stochastic gradient descent are set to 0. Note that the validation AUC for the model with 10 and 15 nodes are approximately equal. However, a neural network with fewer nodes in the hidden layer is considered a simpler model that converges to a solution more quickly.

Another approach to accelerate the stochastic gradient descent is to establish stopping criteria, such as a maximum number of iterations. Table 3.4 displays the AUC values for the neural network model. All the parameters remain fixed except for the maximum iteration count. As indicated in the table, the model's performance improves with a higher number of iterations. It is noteworthy that once stochastic gradient descent converges (in this case, after 610 iterations), the AUC values do not exhibit further improvement. Employing early stopping in the stochastic gradient descent algorithm can act as a form of regularization, mitigating over-fitting. However, in this particular problem, the algorithm converges after 610 iterations and does not demonstrate signs of over-fitting the data.

Another approach for regularization of neural networks and preventing over-fitting is to use a hyper-parameter called "decay" in the model. This approach draws inspiration from ridge and lasso regularization methods used in random forests. Setting the decay parameter to ϕ, a fraction ϕ of the units in a layer is randomly removed, while the remaining units scale up to compensate for the missing units. This regularization method helps prevent nodes from becoming overly specialized. Table 3.5 illustrates the results for different decaying

decay	training set AUC	validation set AUC
0.001	0.6815262	0.6791579
0.01	0.8294086	0.7969051
0.1	0.8244184	0.8010579
0.2	0.8211767	0.8024934
0.5	0.8096603	0.7972103

Table 3.5: This table illustrates the AUC values of a single-layer feed-forward neural network with different decaying rates. All the parameters in the model are fixed and decay parameter changes. The best performance is for 0.1 decay rate.

rates. Although the decay rate of 0.2 yields the best AUC on the validation set, the other decay rates also demonstrate similar performance. This indicates that our neural network is stable, as changes in hyper-parameters do not significantly affect the results.

3.2 Performance metrics

The model performance is assessed using three different statistical metrics: 1. misclassification error rate using the confusion matrix, 2. the area under the receiver operating characteristic curve (AUC), 3. the average log-likelihood.

It is important to note that when measuring the misclassification error rate, different cut-points can be considered for predicting a new observation. Typically, when the data is balanced, meaning that the number of observations in each class is equal, a cut-point of 0.5 is commonly used. However, in the dataset used for this study, only 4.272% of the observations belong to Class 1. As a result, a different cut-point is needed to address the class imbalance, more specifically a cut-point of 0.05 is chosen.

Nevertheless, the miss-classification error rate is not a suitable metric for this dataset, because it is highly imbalanced. Therefore, the AUC and log-likelihood test are more relevant here.

The receiver operator characteristic curve is a probability curve that plots the true-positive rate against the false-positive rate, at different threshold values. The true-positive rate is the proportion of the positive class that is correctly classified, and the false-positive rate is the proportion of the negative class that is incorrectly classified. Therefore, the AUC measures the discriminatory power of a classifier. The AUC ranges between 0 to 1 and the higher the AUC value for a classifier is, the better the classifier can distinguish between the positive and negative classes. The AUC=0.5 here shows that the classifier has no predictive power.

The log-likelihood determines the precision of a regression model. The higher the value of the log-likelihood is, the better a model fits a dataset and the more precise results it gives. Note that the log-likelihood test can only be applied here on the logistic regression method. The log-likelihood value for a single model is meaningless. However, it is useful for comparing two or more models. Therefore, by determining the AUC and log-likelihood, the discriminatory power and the precision of a classifier can be assessed.

3.3 Results

In this section, different risk assessment models are compared, and their predictive performances are explained. The dataset consist of 100,000 observations, 60% of observations randomly selected for the training set, 30% for the test set, and 10% for the validation set. Table 3.6 presents the results of 13 models using four machine learning methods: logistic regression, random forests, gradient boosting trees, and feed-forward neural networks. Each method is applied with three sets of input variables: 1. traditional variables, 2. telematic variables, and 3. both traditional and telematic variables.

In the rest of this section, each model is investigated individually, and them compared to others. The comparisons are based on the area under the curve (AUC) metric obtained from the test set. Additionally, the AUC result from the training set is also considered to identify any signs of over-fitting in the models. Overfitting occurs when a model performs well on the training set but poorly on the test set, indicating a lack of generality.

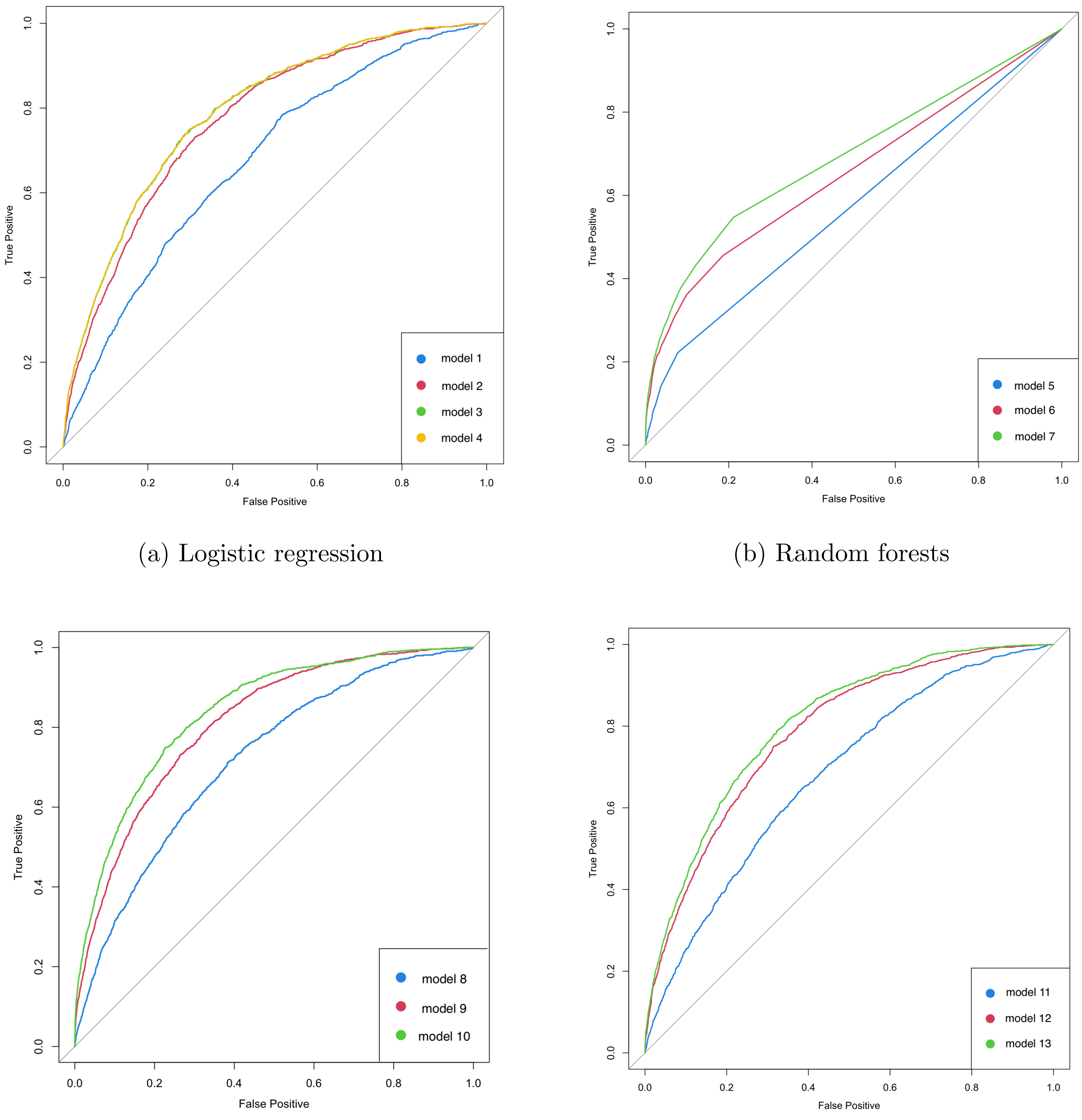
		Miss-classification error rate		AUC		Log-Likelihood	
		test set	train set	test set	train set	test set	train set
Logistic regression	Model 1: traditional predictors	0.3338	0.3348	0.6754	0.6728	-5234.49	-9957.65
	Model 2: telematic predictors	0.2628	0.2618	0.7726	0.7644	-4841.66	-9299.45
	Model 3: traditional+telematic predictors	0.2562	0.2546	0.7888	0.7821	-4733.11	-9105.33
	Model 4: Lasso+traditional+telematic	0.262	0.2582	0.7841	0.7844	-4595.97	-9217.63
Random Forests	Model 5: traditional predictors	0.0464	0.0433	0.5738	0.5468	-Inf	-Inf
	Model 6: telematic predictors	0.0479	0.0464	0.6525	0.6378	-Inf	-Inf
	Model 7: traditional+telematic predictors	0.0467	0.0365	0.6912	0.8467	-Inf	-Inf
Gradient Boosting Trees	Model 8: traditional predictors	0.2938	0.2794	0.7188	0.8114	-4939.79	-8771.72
	Model 9: telematic predictors	0.2277	0.2014	0.8108	0.9383	-4433.75	-6285.37
	Model 10: traditional+telematic predictors	0.217	0.1901	0.8386	0.9526	-4204.51	-5799.34
Feed-Forward Neural Network	Model 11: traditional predictors	0.3173	0.3167	0.6798	0.6872	-5012.84	-10000.74
	Model 12: telematic predictors	0.2699	0.2662	0.7839	0.7887	-4580.28	-9097.48
	Model 13: traditional+telematic predictors	0.2554	0.2507	0.8043	0.8211	-4429.42	-8618.84

Table 3.6: Model performances

First, focusing solely on logistic regression, Model 2, which relies on the telematic variables, demonstrates better performance compared to Model 1, which relies on the traditional variables. However, Model 3, incorporating both data sources, outperforms Models 1 and 2, achieving an AUC of 0.7888. The log-likelihood test is also used to assess the performance of logistic regression models. A higher log-likelihood value indicates better model performance. The log-likelihood results for Models 1, 2, and 3 support the previous findings.

Given the large number of predictive input variables in the dataset, lasso regression is used here to attempt enhancing the out-of-sample prediction accuracy and interpretability. The results reveal that Model 4 exhibits similar performance to Model 3, and lasso regularization does not improve the model. This outcome is due to previous investigation on the VIF of variables, which eliminated correlation between the predictors. Figure 3.2a illustrates the test set ROC curves of Models 1, 2, 3 and 4. Models 3 and 4 exhibit comparable performances. However, the ROC curves of the other two models stand below those of Models 3 and 4.

The prediction performance of the random forests is not as strong as that of other methods employed. Random forests exhibit poor performance on this dataset in comparison to the



(a) Logistic regression

(b) Random forests

(c) Gradient boosting trees

(d) Feed-forward neural network

Figure 3.2: The ROC curves of all 13 models investigated in this study. Each figure presents the models for a specific method. The ROC curve shows false-positive rates on the x-axis and the true-positive rates on the y-axis, and they are computed over the test set.

other methods, likely due to its tendency to overfit. Despite attempts to prune the trees and mitigate over-fitting, random forests do not achieve the desired level of accuracy on the test set. Consequently, random forest is not appropriate for this dataset and it is producing over-fitting. Figure 3.2b illustrates the ROC curves of Models 5, 6 and 7 on the test set.

In addition to logistic regression and random forests, the performances of gradient boosting trees and feed-forward neural networks are also considered. The results of the AUC tests in Table 3.6 indicate that both of these methods exhibit better performance. The neural network method, with both traditional and telematic variables (Model 13), achieves an AUC of 0.8043, and the gradient boosting trees, with both traditional and telematic variables (Model 10), achieves an AUC of 0.8386. These results suggest that these two machine learning methods are capable of capturing non-linear relationships between predictors and the response variables better than logistic regression. Figure 3.3 illustrates the ROC curves for all four methods: logistic regression, random forests, gradient boosting trees and feed-forward neural networks. In this figure, the ROC curves of gradient boosting trees and feed-forward neural networks lies above that of logistic regression (although the lines are close to each other), while the ROC curve of random forests lies lower than that of all other methods, confirming the results presented in Table 3.6.

Furthermore, based on the AUC test results for all four predictive methods presented in Table 3.6, models incorporating telematic data outperform models using only traditional data. However, the best performance is observed when both data sources (traditional and telematic) are included in the model. This finding suggests that including both traditional and telematic data significantly improves the models, indicating that telematic data contains valuable information regarding driving behaviour and accident risk that is not captured by traditional data alone. The results of the log-likelihood test is also consistent with AUC test.

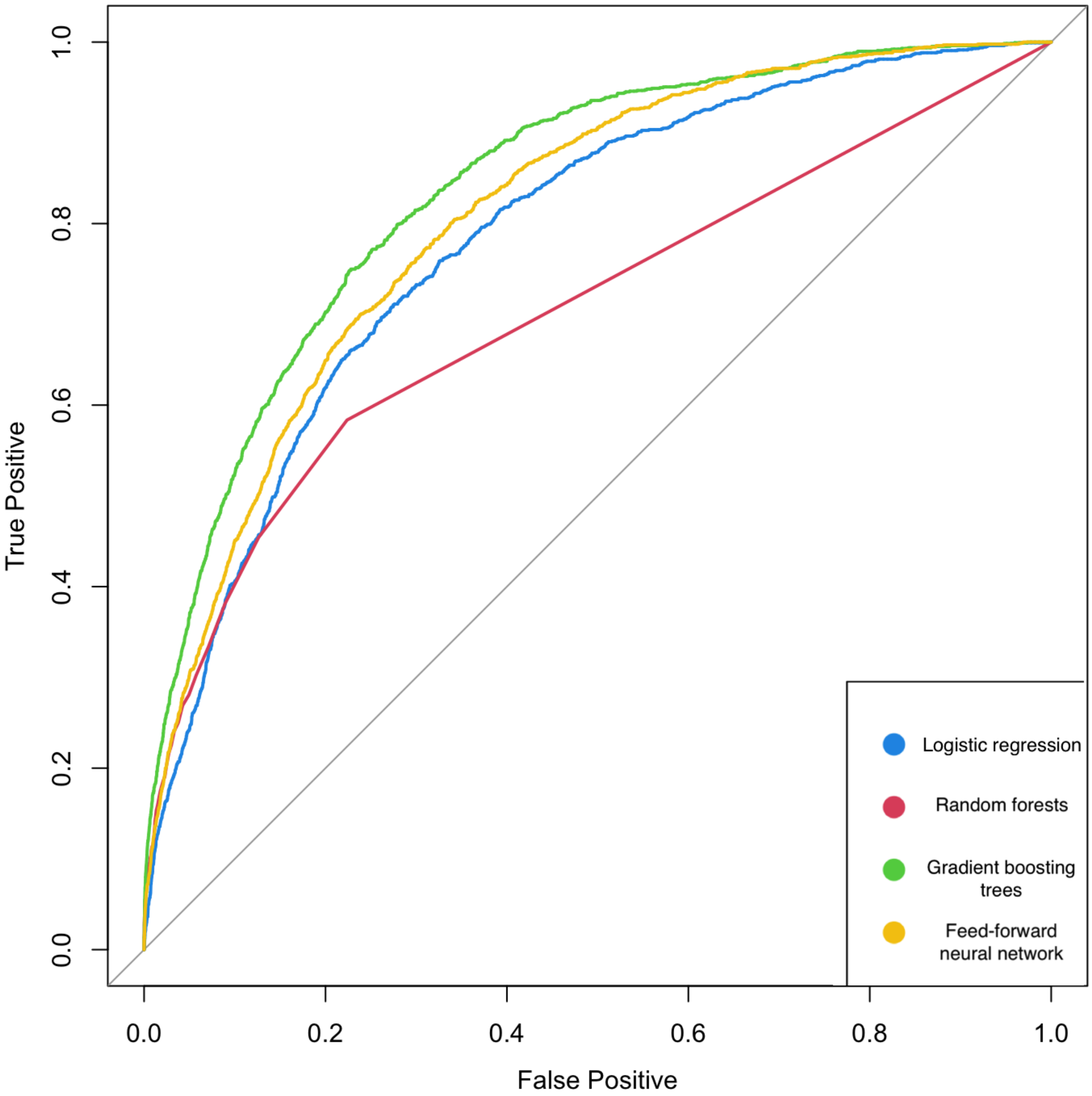


Figure 3.3: This figure illustrates the ROC curves for four methods with all input variables (traditional and telematic). The ROC curves are computed over the test set.

3.4 Variable importance assessment

The risk of having an accident is predicted by combining the informational content of several features. This section aims to provide insights into how each feature contributes to the overall model prediction performance. Thus, one can assess the feature importance in the models.

The assessment of feature importance is carried out using two approaches: (1) Shapley decomposition and (2) marginal performance loss when removing a feature. The Shapley decomposition has recently been introduced in the field of machine learning through algorithms referred to as SHAP (Lundberg and Lee, 2017). These algorithms enable the breakdown of individual predictions into contributions from different features, allowing for the assessment of their respective importance. In this context, the Shapley decomposition quantifies the adjustments made to predictions when subsets of features are augmented with a specific predictor. This decomposition method possesses the valuable property of explaining the contribution of each prediction as the sum of its individual contributions. Therefore, for each observation, the Shapley decomposition provides a comprehensive understanding of how each feature influences the risk probability prediction. In SHAP, the contribution of feature i to the risk probability prediction for observation t is defined as

$$\phi_i = \sum_{S \subset F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{t,S \cup \{i\}}) - f_S(x_{t,S})]$$

where F is the set of all predictors, $|.|$ denotes the cardinality of a set, $x_{t,S}$ is the policyholder t features values for the features subset S, and $f_S(x_{t,S})$ is the risk probability generated by the model trained exclusively with predictors in S. It quantifies adjustments to predictions when the subsets of features are incremented with predictor i. The Shapley decomposition has the favorable property of explaining each prediction contribution as the sum of its contributions. Therefore for every observation t:

$$f_F(x_F) = \phi_{\emptyset,t} + \sum_{i \in F} \phi_{i,t}.$$

To measure the importance of each respective feature, the average absolute feature contributions are presented:

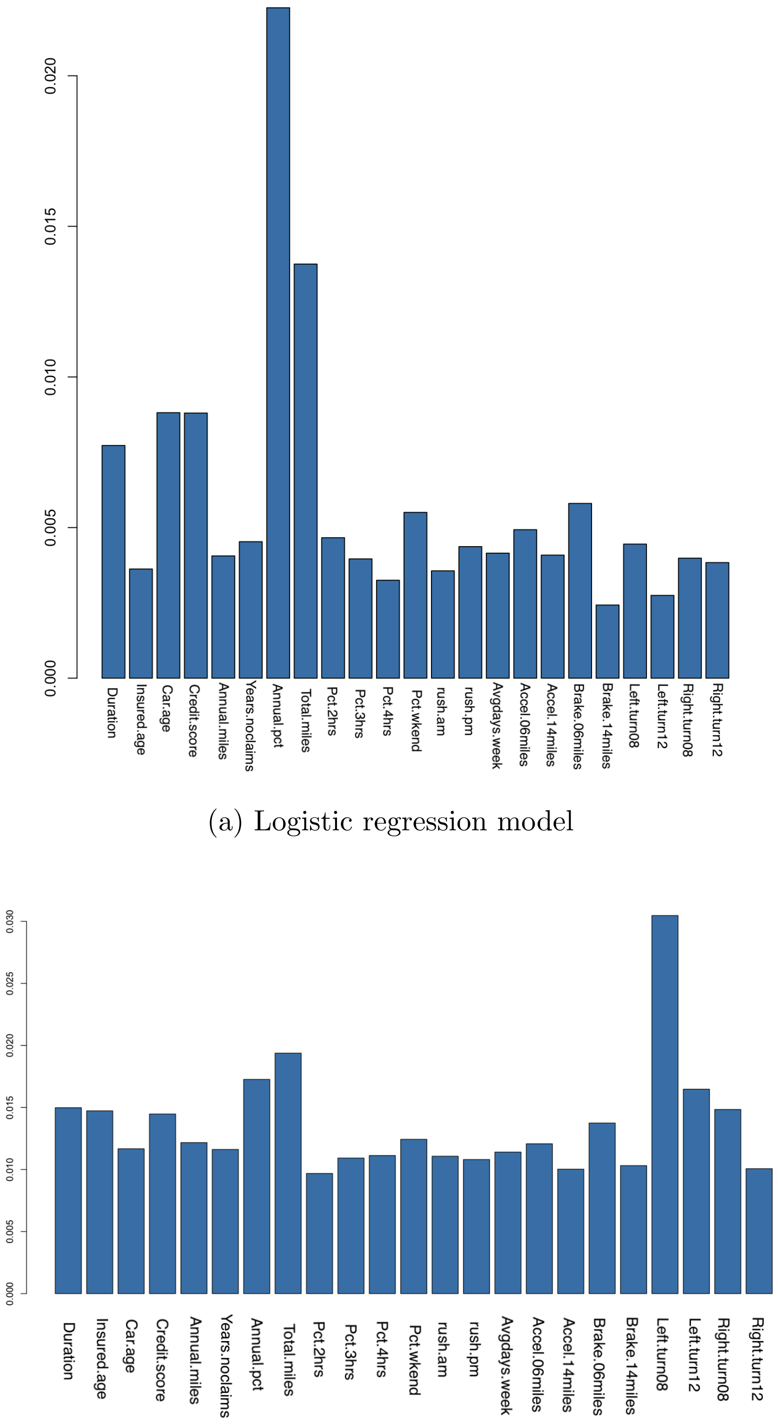
$$\psi_i = \frac{1}{n}\sum_{t=1}^{n} |\phi_{i,t}|,$$

with larger values of ψ_i relative to other features meaning that feature i is more impactful when making predictions. SHAP values are computed using the R package *shapr*. Note that This package is not able to calculate the SHAP values for neural networks.

Figure 3.4 reports the mean absolute Shapley values computed over the out-of-sample observations for every feature. Results indicate that in logistic regression, the "Annual.pct.driven" and "Total.miles.driven" features, and in gradient boosting regression, the "Left.turn.intensity08", "Annual.pct.driven" and "Total.miles.driven" features contribute the most to the predictions. Other features such as "Duration", "Car.age", "Credit.score", "Years.noclaims", "Pct.drive.wkend", "Accel.06miles", and "Brake.06miles" make moderate contributions to the predictions in both models.

To complement the information provided by SHAP, this study quantifies the marginal performance loss through the decreases in out-of-sample average log-likelihood observed when any of the features are omitted from the set during training for logistic regression, gradient boosting trees, and feed-forward neural networks. More precisely, the model is re-trained with a reduced feature set where only the targeted feature is removed, and the ratio of the difference between the out-of-sample log-likelihood from both models (full model minus reduced model) over the log-likelihood of the full model. A drop in performance implies that the feature does bring useful information, while small improvements up to degradation in performance suggest that the feature conveys little to no information. According to Figure 3.5, "Annual.pct.driven" has the highest contribution in logistic regression and feed-forward neural networks models. The "Total.miles.driven" and "Duration" have the highest contribution in gradient boosting regression. Moreover, "Credit.score", "Car.age", "Duration", "Pct.drive.wkend", and "Duration" show moderate contribution to the prediction in all three methods. Such findings are mostly consistent with those provided by the SHAP algorithm.

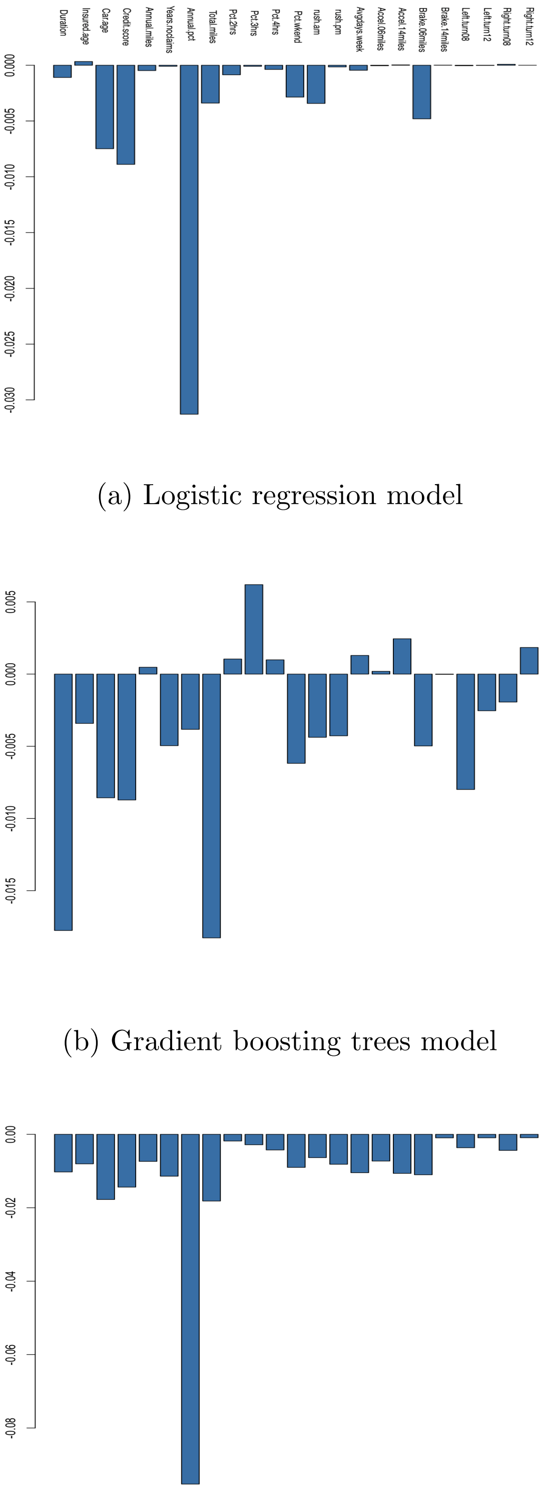
The Shapley decomposition and marginal performance loss provide distinct insights into the contribution of features. The Shapley decomposition measures the reliance of the model



(a) Logistic regression model

(b) Gradient boosting trees model

Figure 3.4: The Shapley values over the out-of-sample set for the logistic regression and gradient boosting trees models.



(a) Logistic regression model

(b) Gradient boosting trees model

(c) Feed-forward neural networks model

Figure 3.5: The percentage decreases in the out-of-sample log-likelihood when a feature is excluded from the feature set, for logistic regression, gradient boosting trees, and feed-forward neural networks.

on individual features, whereas the marginal loss evaluates the incremental performance change when a feature is included in the model. The Shapley decomposition reveals that certain features, which yield minimal or negative marginal performance gains, significantly influence the models. This occurrence arises from the strong interdependence among specific features. For instance, the feature “Left.turn.intensity08” is extensively used by the models according to the Shapley values, despite its negligible associated marginal loss. This outcome can be explained by the fact that other features such as“Left.turn.intensity12” exhibit a strong dependence with“Left.turn.intensity08” and already encompass the relevant information related to predicting accident likelihood.

Conclusion

This research focuses on the analysis of a synthetic car insurance claim dataset that was emulated from a real dataset obtained from a Canadian-based insurer. The dataset comprises 6 traditional variables, 39 telematic variables, and 2 response variables.

An initial examination using a VIF (Variance Inflation Factor) test reveals considerable multicollinearity, particularly among the telematic variables. To address this issue and mitigate variance and overfitting concerns, a subset of variables is selected for the models.

The selected variables for the models consist of 6 traditional variables, 17 telematic variables, and one response variable. Additionally, it should be noted that the dataset used in this study is imbalanced, leading to bias towards the majority class when training machine learning models. Various resampling methods, including oversampling, undersampling, and their combinations, are evaluated, but they do not demonstrate improved prediction performance.

To assess the performance of the models, several metrics are employed, namely the misclassification error rate, AUROC, and log-likelihood. However, it is important to note that the misclassification error rate is not an ideal metric for imbalanced data. Using these metrics, the prediction performance of four machine learning methods is analyzed based on three different predictor sets: (1) traditional data, (2) telematic data, and (3) both traditional and telematic data. The results indicate that gradient boosting trees and one-layer feedforward neural networks exhibit the best performance among all the methods tested. These two machine learning techniques outperform logistic regression method since they capture the underlying non-linear relationships between the predictors and the response variable.

However, it should be noted that logistic regression offers the advantage of being easier to interpret.

On the other hand, the performance of random forests indicates that this particular machine learning method is unsuitable for this dataset, as it exhibits overfitting that is not reduced by regularization and pruning methods.

Furthermore, across all four predictive methods, models that includes both traditional and telematic data demonstrate the best performances compared to models that include only one type of input variables. Additionally, models solely based on telematic data outperform those solely based on traditional data.

Using two feature assessment techniques, namely Shapley decomposition and marginal performance loss through feature removal, indicates that features such as “Annual.pct.driven”, “Total.miles.driven”, ‘Brake.06miles”, “Accel.06miles”, “Pct.drive.wkend”, “Credit.score”,“Car.age”, “Duration”, and “Years.noclaims” influence the likelihood of having an accident the most comparing to other predictors in the dataset.

Bibliography

S. Arumugam and R. Bhargavi. A survey on driving behavior analysis in usage based insurance using big data. *Journal of Big Data*, 6(1):1–21, 2019.

P. Baecke and L. Bocca. The value of vehicle telematics data in insurance risk selection processes. *Decision Support Systems*, 98:69–79, 2017.

L. Barry and A. Charpentier. Personalization as a promise: Can big data change the practice of insurance? *Big Data & Society*, 7, 2020.

C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

J. W. Bolderdijk, J. Knockaert, E. Steg, and E. T. Verhoef. Effects of pay-as-you-drive vehicle insurance on young drivers' speed choice: Results of a dutch field experiment. *Accident Analysis & Prevention*, 43(3):1181–1186, 2011.

J.-P. Boucher and R. Turcotte. A longitudinal analysis of the impact of distance driven on the probability of car accidents. *Risks*, 8(3):91, 2020.

J.-P. Boucher, A. M. Pérez-Marín, and M. Santolino. Pay-as-you-drive insurance: the effect of the kilometers on the risk of accident. In *Anales del Instituto de Actuarios Españoles*, volume 19, pages 135–154. Instituto de Actuarios Españoles, 2013.

J.-P. Boucher, S. Côté, and M. Guillen. Exposure as duration and distance in telematics motor insurance using generalized additive models. *Risks*, 5(4):54, 2017.

L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

M. Denuit and J. Trufin. Effective statistical learning methods for actuaries. 2019.

C. K. Fan and W.-Y. Wang. A comparison of underwriting decision making between telematics-enabled UBI and traditional auto insurance. *Advances in Management and Applied Economics*, 7(1):17, 2017.

A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera. *Learning from imbalanced data sets*, volume 10. Springer, 2018.

G. Gao and M. V. Wüthrich. Convolutional neural network classification of telematics car driving data. *Risks*, 7(1):6, 2019.

G. Gao, S. Meng, and M. V. Wüthrich. Claims frequency modeling using telematics car driving data. *Scandinavian Actuarial Journal*, 2019(2):143–162, 2019a.

G. Gao, M. V. Wüthrich, and H. Yang. Evaluation of driving risk at different speeds. *Insurance: Mathematics and Economics*, 88:108–119, 2019b.

G. Gao, H. Wang, and M. V. Wüthrich. Boosting Poisson regression models with telematics car driving data. *Machine Learning*, pages 1–30, 2021.

M. Guillen, J. P. Nielsen, A. M. Pérez-Marín, and V. Elpidorou. Can automobile insurance telematics predict the risk of near-miss events? *North American Actuarial Journal*, 24(1): 141–152, 2020.

M. Guillen, J. P. Nielsen, and A. M. Pérez-Marín. Near-miss telematics in motor insurance. *Journal of Risk and Insurance*, 2021.

Y. Huang and S. Meng. Automobile insurance classification ratemaking based on telematics driving data. *Decision Support Systems*, 127:113156, 2019.

G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

S. Meng, H. Wang, Y. Shi, and G. Gao. Improving automobile insurance claims frequency prediction with telematics car driving data. *ASTIN Bulletin: The Journal of the IAA*, pages 1–29, 2021.

B. Qi, P. Liu, T. Ji, W. Zhao, and S. Banerjee. Drivaid: Augmenting driving analytics with multi-modal information. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–8. IEEE, 2018.

B. So, J.-P. Boucher, and E. A. Valdez. Synthetic dataset generation of driver telematics. *Risks*, 9(4):58, 2021.

J. Stipancic, L. Miranda-Moreno, and N. Saunier. Vehicle manoeuvers as surrogate safety measures: Extracting data from the GPS-enabled smartphones of regular drivers. *Accident Analysis & Prevention*, 115:160–169, 2018.

S. Sun, J. Bi, M. Guillen, and A. M. Pérez-Marín. Driving risk assessment using near-miss events based on panel Poisson regression and panel negative binomial regression. *Entropy*, 23(7):829, 2021.

M. V. Wüthrich. Covariate selection from telematics car driving data. *European Actuarial Journal*, 7(1):89–108, 2017.

W. Yu, G. Guan, J. Li, Q. Wang, X. Xie, Y. Zhang, Y. Huang, X. Yu, and C. Cui. Claim amount forecasting and pricing of automobile insurance based on the bp neural network. *Complexity*, 2021, 2021.

Appendix A.

In this appendix some visualizations of the dataset are presented. These illustrations can provide insight about the dataset and the possible relationship between the variables. The figures include both traditional and telematic variables visualizations. For example, according to Figure 3.6 younger drivers have a higher chance of having accident, which is consistent with the study of Bolderdijk et al. (2011). Figure 3.7 implies that the drivers with higher credit score have a lower risk of accident, and Figure 3.9 shows the drivers that use their car more frequently have higher risk of accident. Figure 3.7 shows that policyholders with higher score credits have less risk of accident.

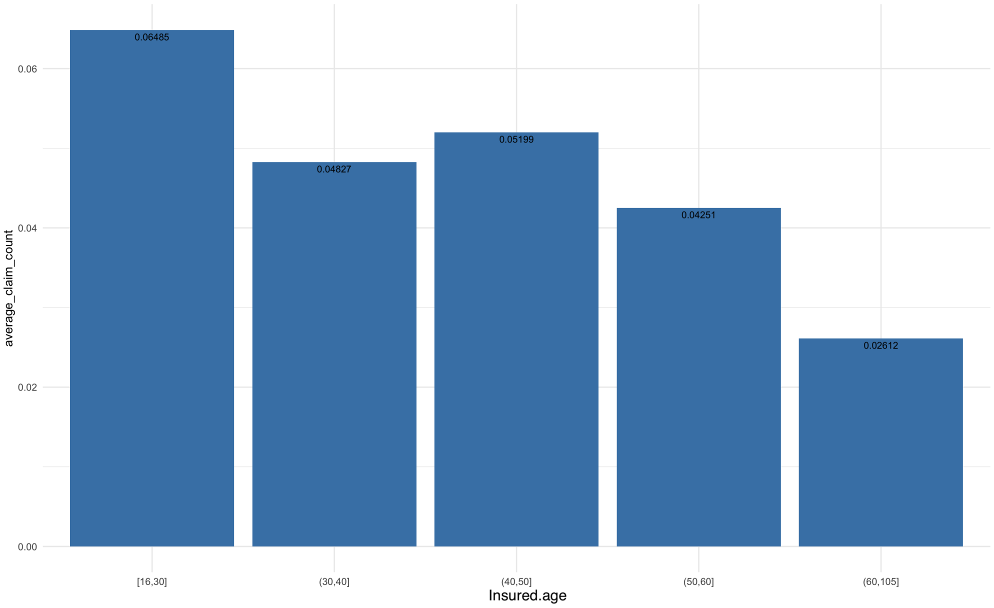


Figure 3.6: The average number of high-risk drivers (drivers with at least one claim during observation) for different age ranges.

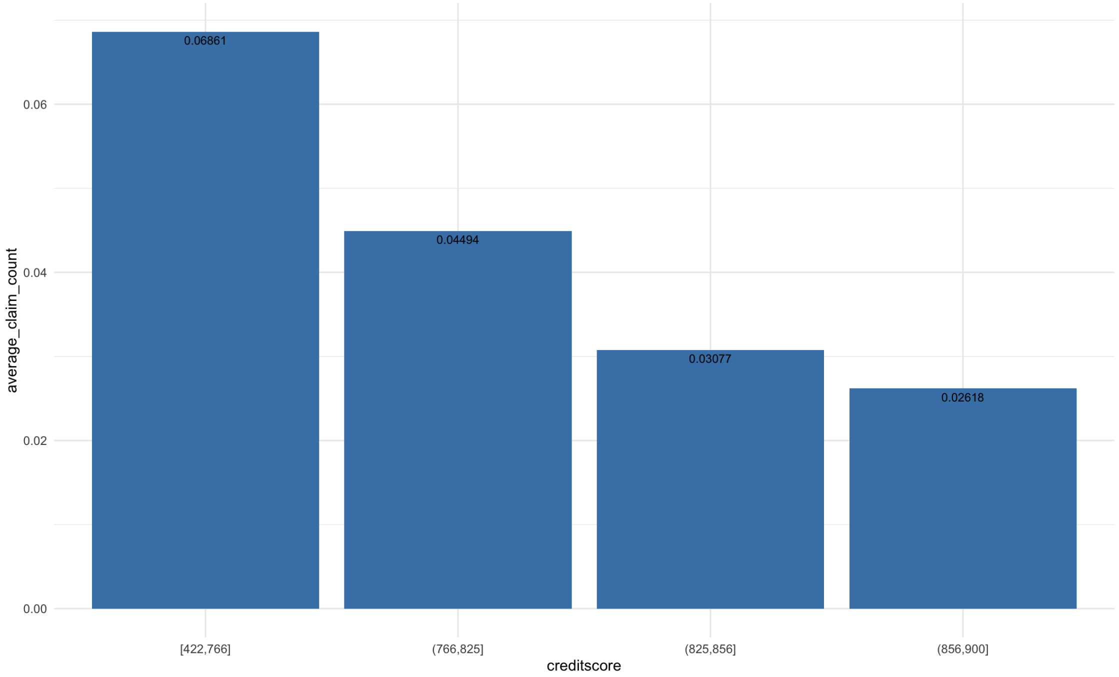


Figure 3.7: The average number of high-risk drivers (drivers with at least one claim during observation) for different credit score ranges.

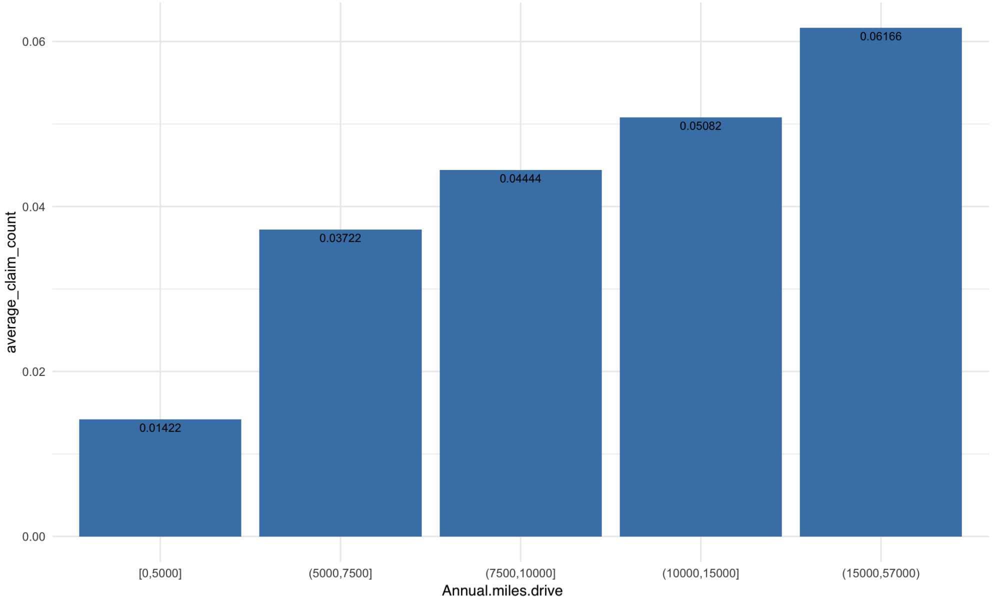


Figure 3.8: The average number of high-risk drivers (drivers with at least one claim during observation) for different declared expected annual distance driven ranging 0-5000 miles, 5000-7500 miles, 7500-10000 miles, 10000-15000 miles and more than 15000 miles .

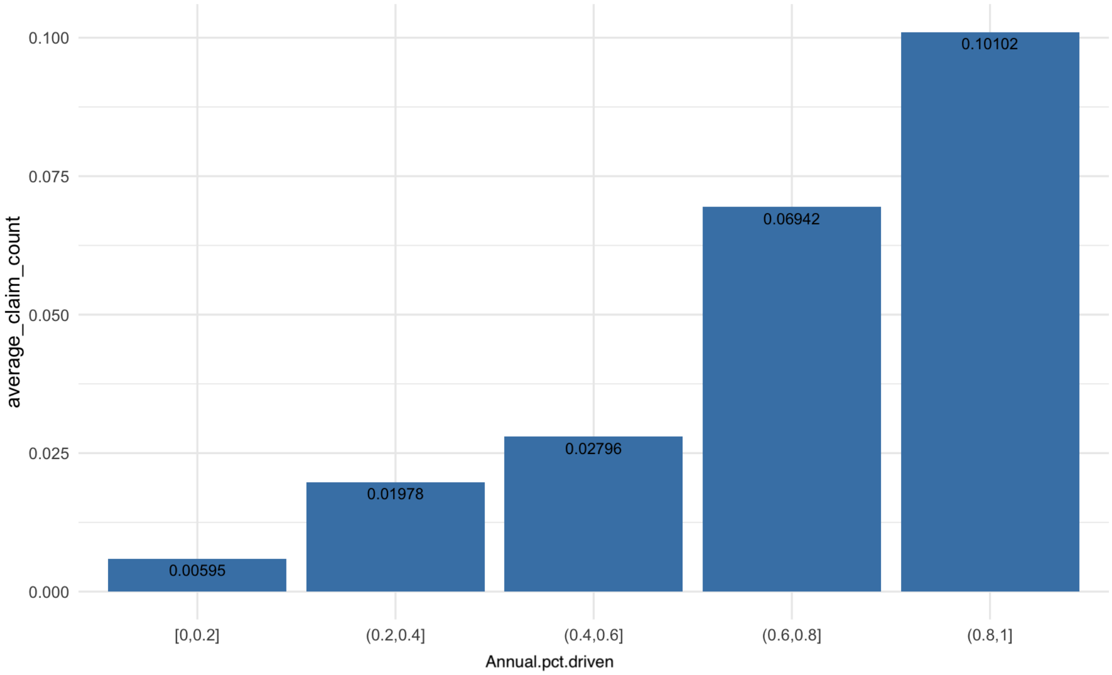


Figure 3.9: The average number of high-risk drivers (drivers with at least one claim during observation) for different annual vehicle use percentage (the number of the day a policy holder uses vehicle divided by 356) ranges.

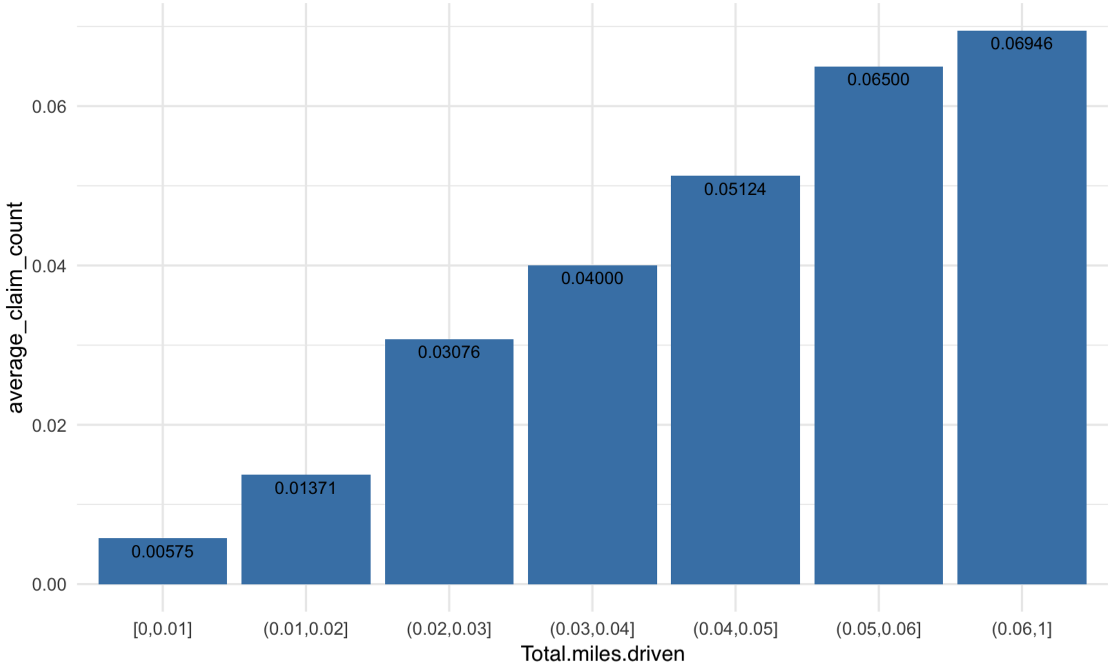


Figure 3.10: The average number of high-risk drivers (drivers with at least one claim during observation) for a given range of total miles driven per day.

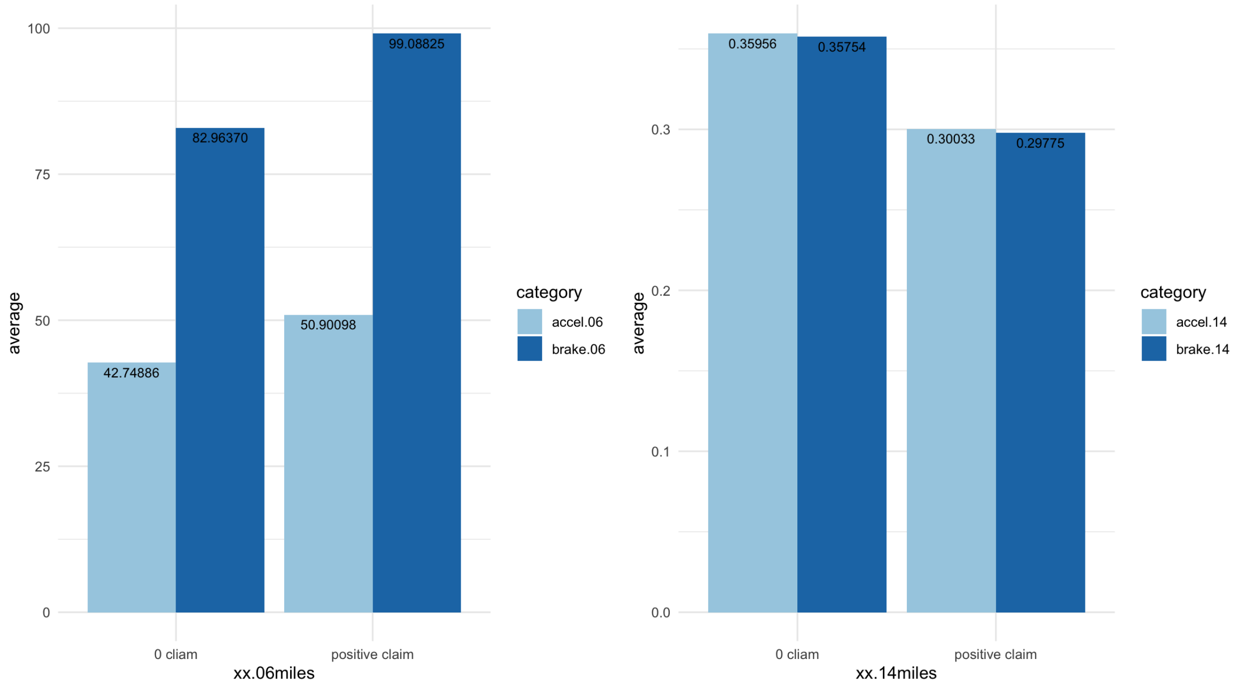


Figure 3.11: The average count of the brakes and accelerations for high-risk drivers (drivers with at least one claim during the observation) and low-risk drivers (drivers without any claim during the observation). The Left panel shows the average count of accelaration (light blue) and brake (dark blue) with intensity equal to 6 mph/s per 1000 miles, and the right panel shows the average count of acceleration and brake with 14 mph/s per 1000 miles.

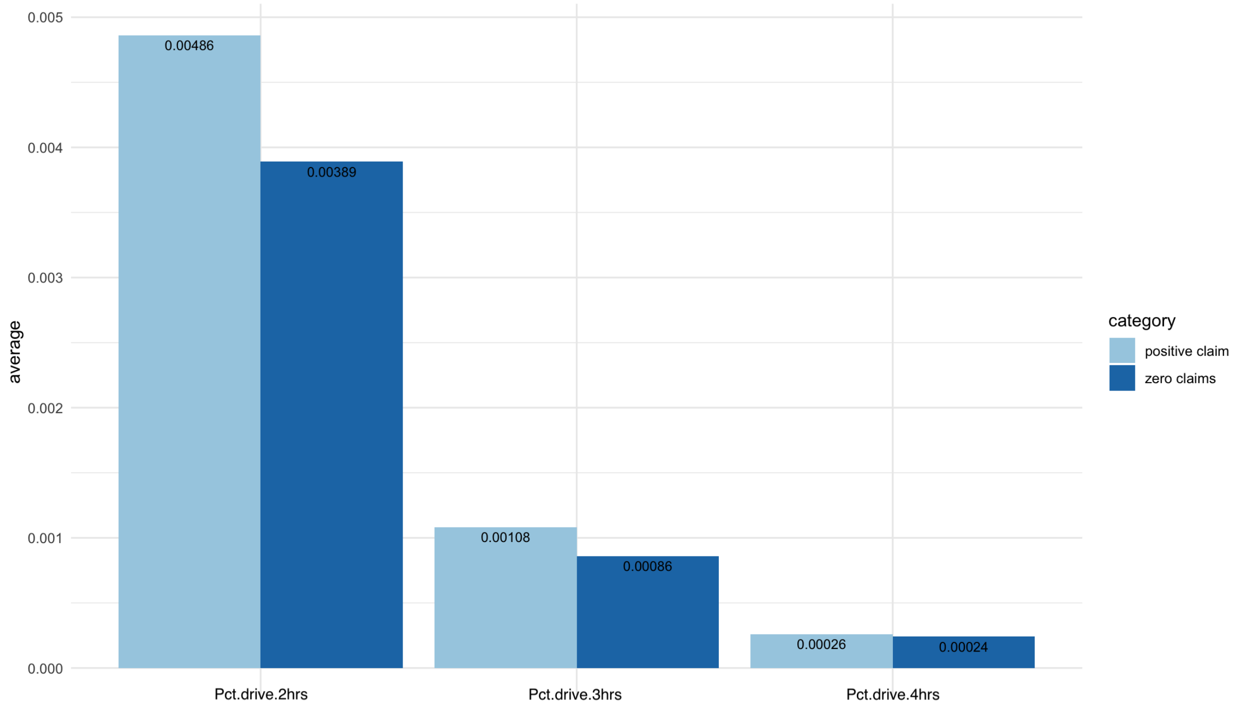


Figure 3.12: The average percentage that the vehicle is driven within 2hrs, 3hrs and 4hrs. The light blue shows the percentages for drivers that have at least one claim during the observation and the dark blue shows the percentages for drivers without any claims during the observation.